

Trabajo de Fin de Máster

Máster Universitario en Ingeniería Industrial

Flexible Job-Shop Problem minimizando los costes en función de la fecha de entrega y el instante de realización

MEMORIA Y ANEXOS

Autor: Oriol Faura Gatus
Director: Rafael Pastor Moreno
Convocatoria: Septiembre 2016



Escola Tècnica Superior
d'Enginyeria Industrial de Barcelona



Resumen

El siguiente Trabajo de Fin de Máster consiste en resolver el problema de programación *flexible Job-Shop* donde el índice de eficiencia a minimizar está formado por la suma de los costes de la energía eléctrica necesaria para ejecutar los trabajos y los costes de adelanto o retraso de los trabajos con respecto a la fecha de entrega de los mismos. Cabe destacar, la búsqueda de una producción *just-in-time* (por la existencia de una fechas de entrega) y la consideración de la variación horaria del precio de la energía eléctrica según el instante de realización (discriminación horaria).

Inicialmente se define la importancia de la programación de operaciones, la clasificación de sus problemas y una amplia descripción del problema de programación de operaciones a solucionar, el cual es considerado uno de los problemas de programación más complejos de resolver de forma óptima. Después de un extenso y detallado estado del arte sobre el *flexible Job-Shop Problem* (fJSP), se llega a la conclusión que el único artículo que considera la discriminación horaria del precio de la energía eléctrica es el presentado por Corominas *et al.* (2015). Este artículo será tomado como referencia para valorar la calidad de las soluciones encontradas con los nuevos procedimientos propuestos.

Se diseñan siete procedimientos distintos para resolver el problema *flexible Job-Shop*. Todos ellos parten de la heurística diseñada por los autores del artículo de referencia, la cual es modificada y mejorada en cada uno de los procedimientos descritos. Los nombres de los siete procedimientos son: Calibración General estática (CGe), Calibración General semidinámica (CGs), Calibración Individual estática (Cie), Calibración Individual semidinámica (CIs), GRASP-HeufJSP (con azar orientado y con azar no orientado), SA-HeufJSP y GRASPSA-HeufJSP.

La calidad de cada uno de los procedimientos se determina comparándolos con los resultados del procedimiento HeufJSP(ζ) de Corominas *et al.* (2015) y, a su vez, entre ellos. Para este fin, se ejecuta un largo experimento computacional, el cual consiste en resolver 3888 ejemplares con cada uno de los procedimientos diseñados. Previamente, se realiza una calibración de los parámetros de los procedimientos con los métodos de Nelder & Mead y CALIBRA.

Con la aplicación de cualquier procedimiento planteado en este trabajo, el resultado que obtendrá el usuario es la programación de las operaciones de todos los trabajos, donde se definirá la máquina asignada a cada operación y su instante de inicio.

Finalmente y con el análisis de los resultados, se enumera el conjunto de beneficios que conlleva la aplicación del procedimiento recomendado y diseñado en este trabajo.

Índice General

1. INTRODUCCIÓN	5
2. OBJETO	6
3. ALCANCE	7
4. PROGRAMACIÓN DE OPERACIONES	8
4.1. Clasificación de los problemas de programación de operaciones	9
4.1.1. Clasificación según la disponibilidad de los trabajos y recursos	10
4.1.2. Clasificación según el tipo de flujo de los trabajos	11
4.1.3. Clasificación según el índice de eficiencia	11
4.2. Planteamiento del problema.....	13
4.2.1. Flexible Job-Shop Problem	13
4.2.2. Nomenclatura.....	14
4.2.3. Objetivos a minimizar	16
4.2.4. Suposiciones del problema	17
5. ESTADO DEL ARTE	19
6. PROCEDIMIENTO DE RESOLUCIÓN	25
6.1. Método de resolución de referencia	26
6.1.1. Heurística HeufJSP	27
6.1.1.1. Fase 1: Obtención de una secuencia de trabajos inicial.....	28
6.1.1.2. Fase 2: Programación óptima de operaciones para cada trabajo	29
6.2. Métodos de resolución propuestos	29
6.2.1. Calibración General estática	30
6.2.2. Calibración General semidinámica.....	32
6.2.3. Calibración Individual estática.....	35
6.2.4. Calibración Individual semidinámica	37
6.2.5. GRASP-HeufJSP	38
6.2.6. SA-HeufJSP	44
6.2.7. GRASPSA-HeufJSP	47
7. EXPERIMENTO COMPUTACIONAL	49
7.1. Diseño de los ejemplares	49
7.2. Calibración de los parámetros.....	52
7.3. Análisis de resultados.....	55
7.3.1. Calibración General	56
7.3.2. Calibración Individual	60

7.3.3. GRASP-HeufJSP	64
7.3.4. SA-HeufJSP	69
7.3.5. GRASPSA-HeufJSP	73
8. RESULTADOS FINALES	77
8.1. Comparativa entre procedimientos	77
8.2. Información resultante de la programación de los trabajos	82
8.3. Beneficios logrados.....	83
9. PLANIFICACIÓN Y COSTES	85
9.1. Planificación del trabajo	85
9.2. Análisis de costes	87
10. IMPACTO MEDIOAMBIENTAL	89
CONCLUSIONES	91
AGRADECIMIENTOS	92
BIBLIOGRAFÍA	93
Referencias bibliográficas	93
Bibliografía complementaria	98
ANEXOS	99
Anexo 1: Grafo de múltiples etapas (HeufJSP).....	99
Paso 1: Construcción del grafo de múltiples etapas.....	99
Paso 2: Cálculo del camino mínimo	101
Anexo 2: Algoritmos de calibración	103
EAGH: Empirically Adjusted Greedy Heuristics.....	103
Nelder & Mead	104
CALIBRA.....	106
Anexo 3: Tablas de resultados.....	107
Calibración General estática	108
Calibración General semidinámica.....	108
Calibración Individual estática	109
Calibración Individual semidinámica	110
GRASP-HeufJSP	111
SA-HeufJSP	112
GRASPSA-HeufJSP	112

1. Introducción

El afán de incrementar la productividad empresarial ha provocado que las industrias manufactureras busquen cualquier oportunidad para poder reducir costes, ganar competitividad y aumentar los beneficios de la compañía. Desde hace ya unas décadas, las empresas se han dado cuenta que una programación de la producción efectiva permite un aumento de la productividad y un mejor aprovechamiento de los recursos disponibles.

La programación de operaciones ha incrementado su grado de importancia en los últimos 25 años. Principalmente, este incremento es debido a un crecimiento de la variedad de la demanda del consumidor, la reducción del ciclo de vida de los productos (especialmente los productos electrónicos), la globalización de los mercados y el desarrollo de nuevas tecnologías de fabricación que proporcionan mayor flexibilidad. Estas presiones del mercado remarcan la necesidad de reducir al mínimo el stock, manteniendo la satisfacción del cliente y los períodos de entrega. En definitiva, para conseguir estas metas hay que encontrar una programación de la producción y de sus operaciones que sea eficiente, efectiva y precisa.

Multitud de autores han diseñado algoritmos que resuelvan la programación de operaciones. Teniendo en cuenta un sistema productivo orientado a procesos, el problema clásico es el *Job-Shop*. Se trata de un problema de optimización combinatoria de gran dificultad para ser resuelto de forma exacta. En este Trabajo de Fin de Máster se considera una variante más amplia del problema y, a su vez, de mayor complejidad y realismo. El problema del presente trabajo se denomina *flexible Job-Shop Problem* (fJSP).

La imposibilidad práctica de encontrar soluciones óptimas debido a la falta de recursos de tiempo de computación, ha conllevado que se investiguen métodos heurísticos y metaheurísticos. Son procedimientos que no aseguran una solución óptima, pero que pueden encontrar una solución factible en un tiempo razonable, teniendo en cuenta la alta frecuencia en que se acostumbra a realizar una programación de operaciones en la industria.

Otra dificultad añadida a este tipo de problemas es la necesidad de adaptar los procedimientos diseñados al caso que se esté estudiando, ya que las particularidades de cada sector e incluso de cada empresa perteneciente a un mismo sector son muy grandes. Es por este motivo que existen distintas variantes del problema *flexible Job-Shop*, en función de las suposiciones consideradas o los objetivos a alcanzar.

Este trabajo toma como punto de partida el artículo redactado por Corominas *et al.* (2015). Se diseñan nuevos procedimientos sobre la misma variante del problema *flexible Job-Shop* y se compara la calidad de las soluciones encontradas.

2. Objeto

El objetivo principal de este trabajo es el diseño de algún procedimiento que resuelva el problema *flexible Job-Shop* y mejore la calidad de las soluciones encontradas en el artículo presentado por Corominas *et al.* (2015), considerado como de referencia de aquí en adelante.

La calidad de las soluciones se mide mediante un índice de eficiencia compuesto que tiene el objetivo de minimizar los costes relacionados con la programación de las operaciones. Los costes que se tienen en cuenta son los costes de la energía eléctrica necesaria para producir los trabajos (en concreto, las operaciones que los forman) y los costes del adelanto y/o retraso respecto a la fecha de entrega de cada trabajo.

Para la minimización del coste de la energía eléctrica se fijan dos metas secundarias: reducir el consumo energético de la producción de los trabajos y priorizar la producción en horas valle. Para este último subobjetivo, es necesario considerar la discriminación horaria del coste de la energía eléctrica en la programación. En el lado de los costes de adelanto y/o retraso, también se marcan los subobjetivos de reducir el stock de producto acabado y evitar los retrasos en la entrega de los productos al cliente. En definitiva, se quiere fomentar la producción *just-in-time*, la cual permite reducir los costes relacionados con el almacenaje de producto acabado y las penalizaciones por superar la fecha de entrega al cliente.

Desde el punto de vista de los procedimientos diseñados, se define la meta de encontrar un método de resolución del problema que mejore la calidad del procedimiento de referencia sin superar la hora de cálculo, establecida como el límite máximo deseado para encontrar una solución.

3. Alcance

En primer lugar, se definen y clasifican los problemas de programación de operaciones según la disponibilidad de recursos, el tipo de flujo o el índice de eficiencia objetivo. También se presenta y describe el problema a resolver de forma detallada, así como la nomenclatura utilizada en el mismo.

Acto seguido, se realiza un amplio estudio del estado del arte sobre el problema *flexible Job-Shop*. Se puede adelantar una división en cinco grandes bloques: ejemplos de aplicación del problema, procedimientos para resolver el problema según la literatura, enfoques que se puede tomar a la hora de diseñar un método de resolución, índices de eficiencia más frecuentes y literatura relacionada con la minimización del consumo de energía eléctrica.

Con la información obtenida en el estudio del estado del arte, se procede al diseño de los siete procedimientos presentados en este trabajo y con los cuales se intentará obtener una mejor calidad de las soluciones. Para cada uno de los procedimientos se detalla de forma precisa su estructura básica.

Para la evaluación de la calidad de los procedimientos diseñados con respecto al procedimiento descrito en Corominas *et al.* (2015), se ejecuta un extenso experimento computacional formado por la resolución de 3888 ejemplares para cada procedimiento. Una vez obtenidos los resultados, se utilizan distintas métricas para comprobar la calidad de cada procedimiento. Previamente y para los procedimientos diseñados que lo necesiten, se realiza un proceso de calibración de los parámetros internos de la estructura de cada procedimiento.

Finalmente, se realiza una comparación entre todos los procedimientos diseñados para poder determinar qué método es el más recomendable a utilizar. Es decir, qué procedimiento aporta un mayor ahorro de costes y/o un mejor comportamiento al resolver ejemplares de características distintas.

No entra en el alcance de la memoria de este Trabajo de Fin de Máster la descripción detallada del código de programación que se ha empleado para diseñar cada uno de los procedimientos. Los códigos, los cuales han sido programados en el lenguaje Java, se proporcionan almacenados en el material audiovisual adjuntado en este trabajo.

No es parte de este trabajo el diseño de una interfaz para que el usuario pueda interactuar con los procedimientos diseñados. Este Trabajo de Fin de Máster tiene la finalidad de desarrollar nuevos procedimientos que mejoren a los actuales para la resolución del problema.

4. Programación de operaciones

La planificación es uno de los procesos claves para la gestión y el control de la producción, siendo un pilar fundamental dentro del mundo de la organización industrial. Se define como el proceso que transforma las previsiones de la demanda en decisiones sobre la producción. Según el horizonte de afectación de las decisiones se distinguen distintos tipos de planificaciones.

La pirámide de jerarquía de planes muestra los diferentes tipos de planes que se tienen que realizar en el histórico de una empresa, línea de producción, etc (ver Figura 4.1). En la parte superior se encuentran los planes con un horizonte mayor y en la inferior los que tienen un horizonte de afectación menor. Se trata de 5 planes que se pueden englobar en 3 grandes grupos: planificación de operaciones, planificación de necesidades y programación de operaciones.



Figura 4.1: Pirámide de jerarquía de planes de una empresa.

El problema a resolver en este trabajo hace referencia a la programación de operaciones. Se trata del nivel de decisión a corto plazo y que está presente en el día a día del sistema productivo de la empresa. La finalidad de la programación es la optimización de los recursos disponibles en función de uno o varios objetivos predefinidos, en definitiva, una correcta programación supone una notable ventaja competitiva para la empresa.

La programación de operaciones se define como la asignación de órdenes de trabajo concretas a determinados recursos, que suelen ser personas y/o máquinas, en unos instantes de tiempo concretos. Por lo tanto, para su ejecución es vital conocer la disponibilidad de los recursos necesarios y el tiempo durante el cual se deben o existe la posibilidad de realizar las operaciones necesarias.

La programación de operaciones se puede dividir en tres acciones básicas que la conforman:

- Asignación de cargas: consiste en asignar a cada operación el recurso en el cual se va a ejecutar.
- Secuenciación: para cada recurso, definir el orden en que se realizarán las operaciones asignadas.
- Temporización: determinar los tiempos de inicio y finalización de las operaciones.

La globalización y, a consecuencia, el aumento de la competencia interempresarial ha conllevado una necesidad de incrementar la eficiencia de la producción a fin de conseguir una reducción de costes. En la búsqueda de esta eficiencia, se han diseñado distintas técnicas de resolución de los problemas de programación de operaciones. Se distinguen dos tipos de categorías: métodos exáctos, que proporcionan la solución óptima del problema, pero que acostumbran a requerir largos (y muchas veces inviables) tiempos de cálculo; y métodos heurísticos, que proporcionan una solución aproximada en poco tiempo. En ocasiones, un procedimiento heurístico puede llegar a proporcionar una solución óptima, pero para corroborarlo es necesario una cota que sirva de referencia.

La resolución de problemas relacionados con la programación de operaciones puede llegar a ser todo un reto. En general, se trata de problemas complejos y de gran dificultad. Son problemas combinatorios y dónde el número de soluciones factibles aumenta de forma exponencial. Por ejemplo, para resolver la mayoría de problemas de programación de operaciones en una empresa mediana es prácticamente imposible utilizar un método exácto, ya que el gran número de combinaciones implicaría un tiempo de cálculo inviable para el horizonte de aplicación de esta clase de planificación. Además, es difícil encontrar modelos de uso general, lo que conlleva hacer clasificaciones cada vez más precisas y diseñar métodos adaptados a cada caso concreto.

4.1. Clasificación de los problemas de programación de operaciones

La variedad de problemas de programación de operaciones permite realizar distintas clasificaciones. Conway *et al.* (2003) describieron una notación para clasificar este tipo de problemas. La notación está formada por 4 parámetros: $A / B / C / D$. A continuación, se describe el significado de estos parámetros.

- A. Indica el procedimiento de llegada de los trabajos. Si los problemas son estáticos, representa el número de trabajos a programar. Puede representarse como n trabajos, siendo n un número arbitrario, pero finito. En cambio, si los problemas son dinámicos, hace referencia a la distribución de probabilidad entre los tiempos de llegada de los trabajos. Las definiciones de problema estático y dinámico se encuentran en el apartado 4.1.1.
- B. Indica el número de máquinas del problema. Se puede definir como m máquinas, considerándose como un número arbitrario y finito.
- C. Describe el tipo de flujo de los trabajos. Los principales flujos son: *Flow-Shop* (F), *Randomly Routed Job-Shop* (R), *Permutation Flow-Shop* (P) y *General* o *Job-Shop* (G). En el caso que se programen trabajos para una sola máquina, este parámetro se omite.
- D. Indica el índice de eficiencia con el cual se evaluará la calidad de la programación efectuada.

En los siguientes apartados se definen los tres principales sistemas de clasificación: según la disponibilidad de los trabajos y recursos, según el tipo de flujo de los trabajos y según el índice de eficiencia a tener en cuenta.

4.1.1. Clasificación según la disponibilidad de los trabajos y recursos

Los problemas de programación de operaciones en máquinas pueden clasificarse en función del momento de disponibilidad de los trabajos y de los recursos. Existen 3 tipos distintos:

- Problemas estáticos: Un número finito de máquinas está disponible desde el instante cero. Además, un número conocido de trabajos llegan de forma simultánea y pueden ser procesados de forma inmediata.
- Problemas semidinámicos: Los instantes de disponibilidad de las máquinas y/o de los trabajos no son los mismos, pero se conocen previamente.
- Problemas dinámicos: Se trata de un problema que representa un proceso continuo. Los trabajos llegan de forma intermitente, en ocasiones se pueden predecir con una distribución de probabilidad, y no se conoce con exactitud su disponibilidad. El horizonte de estudio y el número de trabajos es infinito. El número de máquinas y su disponibilidad inicial es conocida.

4.1.2. Clasificación según el tipo de flujo de los trabajos

Los trabajos necesitan pasar por distintas máquinas para realizarse. El flujo hace referencia al camino o ruta que siguen estos trabajos. Los problemas se suelen clasificar en estos cinco tipos de flujo (uno más que los cuatro definidos por Conway *et al.* (2003)):

- Flow-Shop: Los trabajos siguen de forma secuencial la misma ruta sobre una serie de máquinas. Puede darse el caso que un trabajo no tenga que hacer ninguna operación en alguna máquina. Los trabajos solo pueden pasar una vez por máquina.
- Randomly Routed Job-Shop: En este tipo de flujo, no existe ningún patrón común sobre el movimiento de los trabajos de una máquina a otra. Se trata de un tipo de flujo totalmente contrario al tipo *Flow-Shop*.
- Permutation Flow-Shop: Se trata de un caso particular del tipo de flujo *Flow-Shop*. Los trabajos se pueden ejecutar en cualquier orden, pero es importante que el orden sea el mismo para todas las máquinas.
- Job-Shop: Cada uno de los trabajos sigue un camino distinto a través de las máquinas. Los trabajos pueden tener más de una operación por máquina, es decir, pueden pasar más de una vez por la misma máquina. Las operaciones se tienen que realizar siguiendo una secuencia de precedencias entre ellas, la cual es conocida.
- Open-Shop: Similar al tipo de flujo *Job-Shop*, pero los trabajos no tienen ninguna secuencia de precedencias a seguir. Este hecho significa que las operaciones de un mismo trabajo se pueden realizar en cualquier orden.

4.1.3. Clasificación según el índice de eficiencia

Para resolver un problema de programación de operaciones, es necesario un índice de eficiencia que sirva para conocer la calidad de las distintas soluciones halladas con el fin de poderse quedar con la mejor de ellas. Es por este motivo que los métodos de resolución de problemas de programación de operaciones se clasifican según:

- Índice de eficiencia simple: Se caracteriza por tener un solo objetivo a evaluar.
- Índice de eficiencia compuesto: Se compone por dos o más índices de eficiencia simples relacionados entre sí.

La variedad de índices de eficiencia es muy grande, pero lo más común son los índices

simples. Los clásicos se describen a continuación:

- Minimizar el instante de finalización del último trabajo en ser producido en el sistema (C_{\max}). Este concepto es conocido como *makespan*. El término c_j es el instante de finalización del trabajo j .

$$C_{\max} = \max_j \{c_j\} \quad (\text{Ec. 4.1})$$

- Minimizar el tiempo de permanencia en el sistema del trabajo que mayor tiempo permanece en él (F_{\max}). Siendo F_j el tiempo de permanencia del trabajo j en el sistema.

$$F_{\max} = \max_j \{F_j\} \quad (\text{Ec. 4.2})$$

- Minimizar el tiempo medio de permanencia en el sistema (F_{med}). Siendo F_j el tiempo de permanencia del trabajo j en el sistema y n el número de trabajos total.

$$F_{\text{med}} = \frac{\sum_j F_j}{n} \quad (\text{Ec. 4.3})$$

- Minimizar el tiempo de retraso del trabajo que mayor tiempo de retraso tiene (T_{\max}). Siendo T_j el tiempo de retraso del trabajo j o también conocido como *tardiness*.

$$T_{\max} = \max_j \{T_j\} \quad (\text{Ec. 4.4})$$

- Minimizar el tiempo medio de retraso de los trabajos (T_{med}). Siendo T_j el tiempo de retraso del trabajo j y n número total de trabajos.

$$T_{\text{med}} = \frac{\sum_j T_j}{n} \quad (\text{Ec. 4.5})$$

Finalmente cabe comentar que existen otros índices que cada vez son más usados, como por ejemplo: minimizar el tiempo de adelanto del trabajo que mayor tiempo de adelanto tiene (E_{\max}) o minimizar el tiempo de adelanto medio (E_{med}).

4.2. Planteamiento del problema

El presente Trabajo de Fin de Máster plantea la búsqueda de mejores procedimientos de resolución de una variante del problema clásico de programación de operaciones: el problema del taller mecánico o, en inglés, *Job-Shop Problem* (JSP). El enunciado de este problema es el siguiente (Larrosa y Meseguer, 1998):

“Un conjunto finito de n trabajos tienen que ser producidos en un conjunto finito de m máquinas. Cada trabajo está compuesto por una secuencia conocida de O operaciones, donde cada operación necesita un tiempo de proceso y está asignada a una determinada máquina para ser ejecutada. Todo trabajo tiene una fecha de disponibilidad y otra de entrega entre las cuales debería ser procesado. La solución al problema es encontrar la secuenciación de las operaciones en cada máquina y el intervalo de tiempo durante el cual se procesan, con el objetivo de optimizar un índice de eficiencia.”

En el *Job-Shop Problem*, las operaciones solo se pueden realizar en un tipo de máquinas, es decir, las máquinas solo son útiles para un tipo de funciones muy concretas. Con el fin de estar a la par con las necesidades del mercado de hoy en día, los sistemas productivos tienen que ser más flexibles y eficientes. Para lograr estos objetivos, los sistemas productivos necesitan máquinas automatizadas y flexibles, que sean capaces de realizar múltiples tipos de operaciones (Fattahi *et al.* 2007). El *flexible Job-Shop Problem* (fJSP) es una extensión del *Job-Shop Problem* y tiene en cuenta la flexibilidad de las funcionalidades de las máquinas para realizar diferentes tipos de tareas.

4.2.1. Flexible Job-Shop Problem

La flexibilidad de las máquinas en el *flexible Job-Shop Problem* se demuestra con el hecho que existe la capacidad de que cada operación pueda ser realizada en un subconjunto del total de máquinas disponibles. Por lo tanto, el *Job-Shop Problem* es un caso particular del *flexible Job-Shop Problem*, en que el subconjunto de máquinas que pueden hacer cada operación está formado por una sola máquina.

El hecho que se presente la posibilidad de seleccionar qué máquina realiza cada operación, provoca la aparición de otro proceso de toma de decisión para resolver un problema de programación de este tipo. En concreto, el *flexible Job-Shop Problem* se divide en dos subproblemas:

1. Subproblema de asignación: consiste en asignar cada operación a una máquina que pertenece al subconjunto de máquinas capaces o candidatas a ejecutar dicha

operación.

2. Subproblema de programación: se trata de secuenciar las operaciones asignadas a cada máquina en un orden.

El problema *flexible Job-Shop* es considerado mucho más complejo que el *Job-Shop Problem*. Incluye todas las dificultades y complejidades propias del *Job-Shop Problem* y le falta añadir la dificultad de asignar las operaciones a las máquinas. En definitiva, es un problema *NP-hard* y considerado uno de los problemas de optimización combinatoria más difíciles de resolver de forma exacta (Scrich *et al.* 2004).

Dentro de la clasificación definida en el apartado 4.1, el problema de este Trabajo de Fin de Máster quedaría clasificado como:

$$n / m / G / \text{índice eficiencia compuesto}$$

4.2.2. Nomenclatura

Todo problema se puede caracterizar con terminología, la cual simplifica el significado de los datos y variables del problema. En este apartado se enumera la nomenclatura del *flexible Job-Shop Problem* que se presenta en este trabajo. Los datos necesarios del problema son los siguientes:

- n \equiv Total de trabajos.
- m \equiv Total de máquinas.
- i \equiv Máquina, siendo el intervalo $i = 1, \dots, m$.
- j \equiv Trabajo, siendo el intervalo $j = 1, \dots, n$.
- h_j \equiv Total de operaciones del trabajo j ($j = 1, \dots, n$).
- h \equiv Operación, siendo el intervalo $h = 1, \dots, h_j$.
- O_{jh} \equiv Operación h del trabajo j ($j = 1, \dots, n$)($h = 1, \dots, h_j$).
- M_{jh} \equiv Subconjunto de máquinas en las cuales se puede ejecutar la operación h del trabajo j ($j = 1, \dots, n$)($h = 1, \dots, h_j$).
- P_{ijh} \equiv Tiempo de proceso predeterminado para realizar la operación h del trabajo j en la máquina i ($i = 1, \dots, m$)($j = 1, \dots, n$)($h = 1, \dots, h_j$).
- P_{ijh} \equiv Cantidad de energía necesaria para ejecutar la operación h del trabajo j en la máquina i ($i = 1, \dots, m$)($j = 1, \dots, n$)($h = 1, \dots, h_j$).

- λ_j \equiv Estimación de la energía necesaria para procesar el trabajo j (Ec. 6.4).
- σ_{jh} \equiv Estimación de la energía necesaria para procesar la operación O_{jh} (Ec. 6.5).
- r_j \equiv Fecha (instante) de disponibilidad del trabajo j , es decir, instante a partir del cual se puede empezar el trabajo ($j = 1, \dots, n$).
- d_j \equiv Fecha (instante) de entrega del trabajo j ($j = 1, \dots, n$).
- CI_i \equiv Conjunto de intervalos de tiempo en que la máquina i está ocupada inicialmente ($i = 1, \dots, m$).
- h_v \equiv Tarifa de la energía eléctrica en los intervalos de hora valle.
- h_p \equiv Tarifa de la energía eléctrica en los intervalos de hora punta.
- δ_j \equiv Parámetro que especifica la relación lineal entre el tiempo de adelanto respecto la fecha de entrega y su coste para el trabajo j ($j = 1, \dots, n$).
- β_j \equiv Parámetro de la función cuadrática del coste de retraso, respecto a su fecha de entrega, del trabajo j ($j = 1, \dots, n$).
- γ_j \equiv Parámetro de la función cuadrática del coste de retraso, respecto a su fecha de entrega, del trabajo j ($j = 1, \dots, n$).

Acto seguido se definen las variables de decisión y los resultados del problema:

- t_{jh} \equiv Instante de comienzo de la operación h del trabajo j ($j = 1, \dots, n$) ($h = 1, \dots, h_j$).
- c_j \equiv Instante en que se acaba el trabajo j ($j = 1, \dots, n$).
- i_{jh} \equiv Máquina en la cual se produce la operación h del trabajo j ($j = 1, \dots, n$) ($h = 1, \dots, h_j$) ($i_{jh} \in M_{jh}$).
- T_j \equiv Tiempo de retraso (tardanza o *tardiness* en inglés) respecto a la fecha de entrega d_j del trabajo j ($j = 1, \dots, n$): $T_j = \max(0, c_j - d_j)$.
- E_j \equiv Tiempo de adelanto (precocidad o *earliness* en inglés) respecto a la fecha de entrega d_j del trabajo j ($j = 1, \dots, n$): $E_j = \max(0, d_j - c_j)$.
- $U_{i(jh)jh}$ \equiv Coste de energía requerida para realizar la operación O_{jh} en la máquina i_{jh} .
- CF_i \equiv Conjunto de intervalos de tiempo en que la máquina i está ocupada después de la programación de las operaciones ($i = 1, \dots, m$).
- Ω_{jh} \equiv Proporción del tiempo de proceso de la operación h del trabajo j que se realiza en intervalos de horas valle ($j = 1, \dots, n$) ($h = 1, \dots, h_j$).
- CE_j \equiv Coste de adelanto (precocidad o *earliness*) del trabajo j ($j = 1, \dots, n$).

$CT_j \equiv$ Coste de retraso (tardanza o *tardiness*) del trabajo j ($j = 1, \dots, n$).

$U_j \equiv$ Coste energético necesario para producir el trabajo j ($j = 1, \dots, n$).

4.2.3. Objetivos a minimizar

Anteriormente, se ha clasificado el problema *flexible Job-Shop* a resolver como un problema con índice de eficiencia compuesto. El resultado consiste en obtener una programación factible de las operaciones en las máquinas que minimice la suma de los costes de adelanto y retraso y, además, los costes energéticos para producir las operaciones de los trabajos en las distintas máquinas asignadas.

El objetivo o índice de eficiencia es el mismo que el de Corominas *et al.* (2015). El motivo radica en que dicho artículo servirá como punto de partida y de referencia para los procedimientos de resolución. Por lo tanto, el método de valoración de la calidad de una solución factible será el mismo.

La función objetivo o índice de eficiencia compuesto es definido de la siguiente manera:

$$MIN[z] = \sum_{j=1}^n (CE_j + CT_j + U_j) \quad (\text{Ec. 4.6})$$

El significado y el método de cálculo de los tres términos que conforman la función objetivo del problema es (Corominas *et al.* 2015):

1. Coste de adelanto del trabajo j con respecto a la fecha de entrega d_j . Se considera que el coste sigue una relación lineal con respecto al tiempo de adelanto o precocidad, conocido en inglés como *earliness*.

$$CE_j = \delta_j \cdot E_j \quad (\text{Ec. 4.7})$$

2. Coste de retraso del trabajo j con respecto a la fecha de entrega d_j . Se considera que el coste sigue una relación cuadrática con respecto al tiempo de retraso o tardanza, ampliamente conocido como *tardiness*. Los costes de *earliness* y *tardiness* están relacionados con la voluntad de una producción *just-in-time*.

$$CT_j = \beta_j \cdot T_j^2 + \gamma_j \cdot T_j \quad (\text{Ec. 4.8})$$

3. Coste energético necesario para producir el trabajo j . Este coste tiene en cuenta el

momento del día en que se produce la operación, es decir, se tiene en cuenta la discriminación horaria que existe en la mayoría de las tarifas de las empresas suministradoras de electricidad. Las horas del día se dividen en horas valle, más económicas, y horas punta, de mayor coste.

$$U_j = \sum_{h=1}^{h_j} U_{i(jh),jh} \quad (\text{Ec. 4.9})$$

$$U_{i(jh),jh} = (\Omega_{jh} \cdot hv + (1 - \Omega_{jh}) \cdot hp) \cdot P_{i(jh),jh} \quad (\text{Ec. 4.10})$$

4.2.4. Suposiciones del problema

Es necesario definir una serie de suposiciones que sirvan para definir, simplificar y acotar el problema *flexible Job-Shop* que se desea resolver. A continuación, se enumeran las doce suposiciones del problema que son compartidas con el artículo Corominas *et al.* (2015):

1. Toda máquina i requiere una cantidad predefinida de energía (P_{ijh}) para procesar cada operación O_{jh} .
2. El coste de procesar las operaciones depende de la máquina en la cual se procesa, ya que hay diferencias de la energía necesaria para realizar la operación, y el momento del día en que se ejecute, debido a la discriminación horaria.
3. Para cada trabajo j , la secuencia de operaciones a seguir es fija y conocida.
4. Cada máquina solo puede realizar una operación a la vez.
5. El tiempo está discretizado en intervalos de tiempos regulares. Todos los tiempos que participan en el problema, como la fecha de disponibilidad de los trabajos o la fecha de entrega, son múltiplos de dichos intervalos.
6. Se conocen las ocupaciones iniciales de las máquinas (CI_i).
7. Para cada trabajo j , la fecha de disponibilidad r_j es conocida.
8. No se puede interrumpir una operación una vez sea iniciada.
9. Cada trabajo j debería ser completado en su fecha de entrega d_j , ni antes ni después (*just-in-time*).

10. Los tiempos de preproceso se consideran independientes de la secuencia y se incluyen en el tiempo de proceso de las operaciones.
11. No existen restricciones de precedencia entre operaciones de diferentes trabajos.
12. Se considera que hay un espacio ilimitado de espera entre máquinas.

5. Estado del arte

Los problemas de programación son muy comunes y ocurren siempre que en los sistemas productivos haya una elección, como el orden en que un número de tareas se tienen que realizar o los programas que se tienen que ejecutar en un ordenador. A algunos de estos problemas se les encuentra una secuenciación, de las muchas existentes, gracias a la experiencia o automáticamente sin saber que este tipo de problema existía. No obstante, hay una gran cantidad de problemas en la industria, el transporte, actividades institucionales u otros sectores donde los resultados de la secuencia no son triviales.

Tal y como se ha comentado en anteriores apartados, el problema del taller mecánico o *Job-Shop Problem* es una de las ramas de la programación de la producción, siendo uno de los problemas existentes de optimización combinatoria más complejos. Hay dudas sobre que autor o autores fueron los primeros en plantear el *Job-Shop Problem*, pero es ampliamente aceptado atribuir este hecho a Muth y Thompson (1963) que editaron el libro "Industrial Scheduling". Con el paso del tiempo y la evolución de la economía mundial, las empresas manufactureras han tenido la obligación de aumentar su flexibilidad y eficiencia a fin de obtener una ventaja competitiva respecto a su competencia. El problema *flexible Job-Shop* nace para solucionar y encontrar secuencias de operaciones para este nuevo tipo de empresas. Se trata de una ampliación del problema *Job-Shop Problem*, ya que se presenta la capacidad de realizar una misma operación en un conjunto de máquinas candidatas. Uno de los primeros artículos que abordaron el *flexible Job-Shop Problem* fueron Brucker y Schlie (1990). En este artículo propusieron un algoritmo de tiempo polinomial (*polynomial time algorithm*) para resolver el *flexible Job-Shop Problem* con dos trabajos.

Aunque la literatura relacionada con el problema *flexible Job-Shop* es bastante más escasa comparándola con el *Job-Shop Problem*, muchos han sido los autores que han resuelto y presentado casos industriales específicos, lo cual demuestra la gran importancia y la aplicación directa de este tipo de problema de programación de operaciones en el sistema industrial actual. Tanev *et al.* (2004) presentaba un enfoque para programar los pedidos de los clientes en empresas que se dedicaban a fabricar piezas por inyección de plástico. Buscaba obtener una secuencia eficiente para planificar la asignación de los pedidos en las distintas máquinas de inyección de plástico. Chen *et al.* (2007) propuso un algoritmo de secuenciación para el *flexible Job-Shop Problem* con máquinas en paralelo y un proceso de reentrada, en que un mismo trabajo puede pasar más de una vez por una misma máquina. Utilizó, como caso de estudio, el caso real de una empresa especializada en la producción de armas. Otro ejemplo de aplicación real fue presentado por Li y Huo (2009), en el artículo verificaba sus propuestas de resolución con el sistema de producción de tubos de acero no soldados de la empresa Baoshan Iron & Steel Complex, con sede en Shanghái (China).

Thörnblad *et al.* (2015) diseñaron diferentes algoritmos para resolver el problema específico de la multinacional GKN Aerospace Sweden para su célula multi-tarea. Ésta se componía de cinco máquinas idénticas enlazadas en un sistema de fabricación flexible y que, además, manejaban todos los materiales, herramientas y gestión de la información necesaria. Finalmente, otros casos son: Vaghefinezhad y Wong (2012), que expusieron una solución a la secuenciación de trabajos en una empresa manufacturera de diferentes tipos de válvulas de gas; y Calleja y Pastor (2013), los cuales presentaron un *flexible Job-Shop Problem* perteneciente a una compañía de la industria automovilística dedicada a la fabricación de piezas metálicas.

Desde la aparición de los primeros artículos relacionados con el *flexible Job-Shop Problem*, no han parado de realizarse nuevas investigaciones en las cuales se proponen nuevos métodos y algoritmos para resolver este tipo de problema. Cuando se piensa en resolver un problema, el primer objetivo es conseguir la mejor solución posible, también llamada solución óptima. Multitud de autores han mostrado procedimientos fundamentados en la programación matemática: Gomes *et al.* (2008), Birgin *et al.* (2013) y Thörnblad *et al.* (2015) son claros ejemplos de autores que han propuesto un método exacto de resolución. Los modelos matemáticos son adecuados para problemas pequeños, entendiéndose como un caso en que haya pocas máquinas, pocos trabajos y/o con un número reducido de operaciones a programar. No obstante, cuando los problemas son de un mayor tamaño, siendo éste un caso muy común en la realidad industrial, la programación matemática pierde su capacidad de resolución y se convierte en prácticamente inviable. La necesidad de recursos, especialmente tiempo, es tan enormemente grande que convierte el método en impracticable en la realidad. Se debe recordar que la programación de operaciones es la planificación con un horizonte menor (1 a 6 semanas) y no se puede invertir un tiempo excesivo en encontrar una secuenciación óptima. En estos casos, es mejor encontrar una buena solución en un tiempo aceptable que otra muy buena u óptima en un tiempo desproporcionado. Y es en este sentido dónde la gran mayoría de autores coinciden y proponen distintos métodos heurísticos para resolver el problema *flexible Job-Shop*.

Chen *et al.* (2007) y Calleja y Pastor (2013) propusieron algoritmos conocidos como *dispatching algorithm*. Históricamente, la gran mayoría de autores se han decantado por procedimientos metaheurísticos. Uno de ellos es el algoritmo *Tabu Search* (TS) que ha sido planteado, entre otros, por Dauzère-Pérès y Paulli (1997), Saidi-Mehrabad y Fattahi (2006) o, más recientemente, por Vilcot y Billaut (2011). Najid *et al.* (2002) y Fattahi *et al.* (2009) formularon una metaheurística *Simulated Annealing* (SA). Los algoritmos genéticos o *Genetic algorithm* (GA) son uno de los más utilizados, sobre todo combinándolo con otras metaheurísticas, tal y como se verá más adelante. Dos ejemplos claros de aplicación en solitario son los artículos de Gen *et al.* (2009) y De Giovanni y Pezzella (2010). Otro procedimiento clásico es el algoritmo *Variable Neighborhood Search* (VNS) que fue utilizado

por Bagheri y Zandieh (2011).

Durante la última década ha aumentado el número de investigaciones sobre el *flexible Job-Shop Problem*, lo que ha conllevado la aparición de artículos que exponen metaheurísticas más complejas y menos frecuentes. Zarrouk *et al.* (2015) y Singh y Mahapatra (2016) propusieron una metaheurística *Particle Swarm Optimization* (PSO) para resolver el problema. Otro ejemplo es el algoritmo *Ant Colony Optimization* (ACO), el cual fue expuesto en los artículos presentados por los autores Rossi y Dini (2007) y Huang *et al.* (2013). Thammano y Phu-ang (2013) formularon la metaheurística *Artificial Bee Colony* (ABC), un procedimiento muy poco frecuente en la literatura de este tipo de problema de programación. Otro caso similar es el propuesto por Bagheri *et al.* (2010), en su artículo planteó un *Artificial Immune Algorithm* (AIA). Finalmente, Chiang y Lin (2013) propusieron una variante del *Evolutionary algorithm* (EA).

Se han presentado distintos métodos para resolver el *flexible Job-Shop Problem* basados en algoritmos puros. Sin embargo, la bibliografía muestra gran cantidad de artículos dónde se presentan procedimientos híbridos, que son la combinación de dos o más algoritmos puros. Una de las combinaciones más habituales son los *dispatching algorithms* con algún tipo de metaheurística. Por ejemplo: Tanev *et al.* (2004) desarrolló un algoritmo híbrido que juntaba *dispatching algorithm* con *Genetic Algorithm* y, por otro lado, Scrich *et al.* (2004) enseñó cómo combinar *dispatching algorithm* más *Tabu Search*. Xia y Wu (2005) publicaron un artículo dónde describían un algoritmo híbrido juntando *Particle Swarm Optimization* con *Simulated Annealing*. Años más tarde, Roshanaei *et al.* (2013) introdujo un programa matemático para obtener soluciones óptimas en problemas pequeños. Para poder resolver los problemas de gran envergadura, acopló los algoritmos *Artificial Immune* y *Simulated Annealing* (AISA). Ese mismo año, fueron Shahsavari-Pour *et al.* (2013) los que presentaron una original metaheurística dónde hibridó *Genetic Algorithm* y *Simulated Annealing* (NHGASA). Li y Gao (2016) anunciaron otra mezcla entre dos metaheurísticas, siendo esta vez *Tabu Search* más *Genetic Algorithm*.

El problema *flexible Job-Shop* se puede dividir en dos partes: subproblema de asignación, que consiste en asignar cada operación a una máquina perteneciente al grupo de máquinas capacitadas, y el subproblema de programación, dónde se secuencian las operaciones asignadas a cada máquina en un orden que optimicen un índice de eficiencia predefinido y deseado. La literatura de este problema muestra dos enfoques a la hora de intentar resolverlo, se trata del enfoque jerárquico y el enfoque integrado. Por un lado, el enfoque jerárquico se fundamenta en considerar de forma separada la asignación de las operaciones en las máquinas y la programación de las operaciones en las propias máquinas. Por otro lado, el enfoque integrado consiste en resolver tanto la asignación como la programación a la vez, en una sola etapa. El enfoque jerárquico tiene el objetivo de simplificar la resolución del problema. Cabe recordar que en el problema clásico *Job-Shop*, las operaciones ya

tienen definidas de forma implícita la máquina en que se va a llevar a cabo y “sólo” hay que definir la secuenciación. Es decir, resolviendo primero la asignación de las operaciones, la parte que queda es un *Job-Shop Problem*. Brandimarte (1993) fue el primero en utilizar el enfoque jerárquico para resolver el *flexible Job-Shop Problem*. Solucionó el subproblema de asignación con *dispatching algorithm* y luego se centró en el subproblema de programación con una metaheurística *Tabu Search*. Saidi-Mehrabad y Fattahi (2006) propusieron otro *Tabu Search* descompuesto en dos partes para resolver este problema de programación. Fattahi *et al.* (2009) presentó un artículo que resolvía el problema mediante una metaheurística *Simulated Annealing* con enfoque jerárquico. Otros ejemplos más de procedimientos jerárquicos son, por ejemplo, De Giovanni y Pezzella (2010) y Ishikawa *et al.* (2015). Desde el punto de vista del enfoque integrado, éste fue aplicado en la obra de Hurink *et al.* (1994), en la cual sugirió utilizar un *Tabu Search*. Otros autores en considerar esta misma metaheurística con un enfoque integrado fueron Dauzère-Pérès y Paulli (1997). Mastrolilli y Gambardella (2000) presentaron dos procedimientos de optimización local para mejorar los resultados de Dauzère-Pérès y Paulli. González *et al.* (2015) diseñó un algoritmo *Scatter Search* con *Path Relinking* (SSPR) con enfoque integrado. Otros artículos que han aplicado un enfoque integrado en sus procedimientos para resolver el problema son: Bagheri *et al.* (2010); Roshanaei *et al.* (2013) y Thammano y Phu-ang (2013), entre otros.

Hasta ahora se han presentado distintos procedimientos para poder resolver el problema *flexible Job-Shop*, como también dos tipos de enfoques antagónicos. Todos los procedimientos anteriormente enumerados son válidos para obtener una solución final, pero se necesita obtener una evaluación de las soluciones para poderlas comparar entre sí y escoger así el mejor procedimiento de resolución. Lo mismo sucede cuando con un mismo procedimiento se obtienen múltiples soluciones, hecho que se da, por ejemplo, con las metaheurísticas. La evaluación de las soluciones se obtiene calculando un índice de eficiencia predefinido en función de los requerimientos u objetivos que se marcan. Uno de los índices de eficiencia por excelencia es minimizar el momento o tiempo en que se termina el último de los trabajos a programar, en inglés es nombrado como *makespan*. Este índice se utiliza con el objetivo de compactar al máximo la programación de operaciones y minimizar así los tiempos ociosos de los recursos, que normalmente son considerados como máquinas. Rossi y Dini (2007) o Gen *et al.* (2009) son algunos autores que utilizaron el *makespan* como el índice de eficiencia de sus respectivos artículos. Hay otros investigadores, como por ejemplo Scrich *et al.* (2004), que definieron un procedimiento donde su objetivo a minimizar era la suma de las tardanzas o retrasos en la finalización de los trabajos con respecto a una fecha de entrega o vencimiento. Este concepto es conocido como *tardiness* y es utilizado en aquellos casos en que es muy importante cumplir con las fechas pactadas con los clientes, ya sea por motivos estratégicos o por penalizaciones contractuales. El paso del tiempo ha llevado a utilizar índices de eficiencia compuestos, los cuales están formados por dos o más índices simples. Uno de ellos es el que combina el

concepto *tardiness* con el *earliness*, siendo éste la precocidad en que termina un trabajo con respecto a la fecha de entrega. Las empresas se han dado cuenta que todo lo que tienen en los almacenes tiene y supone un coste y, por lo tanto, la combinación de estos índices busca terminar los trabajos sin adelanto ni retraso. Dos obras que combinan estos objetivos son las de Wu y Weng (2005) y Huang *et al.* (2013). Existen otros autores que han diseñado índices de eficiencia compuestos con tres o más índices. Tay y Ho (2008) presentaron tres índices a minimizar: *makespan*, promedio de *tardiness* y promedio del tiempo requerido para completar los trabajos, también conocido como *flow time*. Li *et al.* (2012) y Shahsavari-Pour *et al.* (2013) aplicaron el mismo índice compuesto basado en minimizar el *makespan*, la carga de trabajo de la máquina más sobrecargada y la carga de trabajo total de todas las máquinas.

La globalización y el aumento del consumo han conllevado que el sector industrial sea uno de los mayores consumidores de energía eléctrica. La disminución de los recursos de combustibles fósiles y su aún dependencia para la producción de energía eléctrica ha provocado que el precio del kilowatt hora (kWh) haya aumentado de forma sustancial. Este hecho ha provocado que las empresas intenten optimizar sus recursos para disminuir su coste de producción y ser así más eficientes. Las investigaciones siempre van en paralelo con las necesidades y fue Mouzon (2008) quien publicó una completa disertación, en la cual se tenía en cuenta la reducción del consumo de energía eléctrica. Introdujo el concepto de ahorro de energía mediante la parada y puesta en marcha de la máquina entre operaciones. Definía que existen situaciones entre operaciones de una misma máquina, en que es más eficiente pararla y volverla a encender que no tenerla en modo *stand-by*. Jiang *et al.* (2014) planteó una metaheurística *Genetic Algorithm* con una función multiobjetivo que incluía minimizar la energía consumida durante la producción. En el mismo año, Liu *et al.* (2014) presentaron un artículo para resolver el *Job-Shop Problem* con un índice de eficiencia compuesto, en el cual se minimizaba el consumo total de energía eléctrica y el *tardiness*. Consideraba una distinción de la energía en tres tipos: energía de *stand-by*, energía de funcionamiento (sin elaborar la operación) y energía de ejecución (elaborando la operación). He *et al.* (2015) formularon la misma distinción de la energía eléctrica. En este caso, el problema a resolver era un *flexible Job-Shop*. Otros autores que consideraron como índice de eficiencia el consumo eléctrico fueron Firas y Denis (2015). Establecieron en su artículo la suposición que toda máquina sigue la siguiente curva de potencia-tiempo: proceso de puesta en marcha de la máquina, régimen estacionario en *stand-by*, transitorio entre el régimen de *stand-by* y ejecución y, finalmente, régimen estacionario de ejecución de la operación. Song *et al.* (2014) y Liu y Tiwari (2015) son otros investigadores que han incluido minimizar el consumo de energía eléctrica en sus respectivos índices de eficiencia.

Los artículos citados hasta ahora tienen en cuenta la minimización del consumo de energía eléctrica, pero no están directamente relacionados con minimizar el coste de energía

eléctrica real. Actualmente, las compañías suministradoras de energía tienen tarifas que relacionan el coste energético con el concepto de discriminación horaria, es decir, la energía no cuesta lo mismo en todas las horas del día. Es aquí donde aparecen las horas valle, menor coste de la energía, y las horas punta, mayor coste. En términos de coste, para una empresa es más interesante producir en horas valle, ya que el coste de producción será menor. Si además se tiene en cuenta la diferencia de energía entre máquinas para ejecutar una misma operación, se llega a una minimización doble del coste energético: seleccionando las horas de menor coste eléctrico y las mejores máquinas con respecto al consumo. El único artículo encontrado en el momento de realizar el presente estado del arte, que tenga en cuenta la discriminación horaria y resuelva el *flexible Job-Shop Problem*, es el propuesto por Corominas *et al.* (2015). Este Trabajo de Fin de Máster tomará como base este artículo para el diseño de nuevos procedimientos que mejoren la calidad de las soluciones encontradas.

6. Procedimiento de resolución

En este apartado, se presentarán los distintos algoritmos o procedimientos diseñados para resolver el problema *flexible Job-Shop*, el cual ha sido definido en el apartado 4.2 y sus respectivos subapartados. Cabe recordar que el índice de eficiencia es de tipo compuesto, ya que está formado por tres costes a minimizar que hacen referencia a distintos conceptos: *tardiness* o tardanza respecto a la fecha de entrega; *earliness* o precocidad respecto a la fecha de entrega; y coste energético para producir los trabajos, teniendo en cuenta el concepto de discriminación horaria y la diferencia entre máquinas.

Los dos primeros costes tienen la finalidad de buscar una programación de los trabajos que tenga una fecha de finalización lo más cercana posible a la fecha de entrega o de vencimiento. A este tipo de producción se la denomina *just-in-time*, realizar los trabajos en el momento justo. Con esto se consigue evitar los stocks de producto acabado y también las posibles penalizaciones que pueda conllevar un retraso en la entrega de los trabajos a los clientes.

Este Trabajo de Fin de Máster tomará como referencia el procedimiento y los resultados obtenidos en el artículo publicado por Corominas *et al.* (2015). El objetivo es diseñar métodos resolutivos que mejoren la calidad mostrada en las investigaciones presentadas en el artículo de referencia. Tal y como se ha comentado en apartados anteriores, se trata del único trabajo encontrado que tiene en cuenta el concepto de discriminación horaria para la resolución del problema *flexible Job-Shop*.

Se proponen siete procedimientos de resolución distintos para resolver el problema. Los dos primeros, Calibración General estática y Calibración General semidinámica, parten de la heurística esencial del método de referencia, la cual se denomina HeufJSP (Corominas *et al.* 2015), y después de realizar un proceso de calibrado de sus respectivos parámetros, se llega a obtener soluciones con gran rapidez. Los procesos de Calibración Individual estática y Calibración Individual semidinámica calibran sus respectivos parámetros adaptándolos al ejemplar del problema que se resuelve y, por lo tanto, no son comunes a todos los ejemplares, tal y como sucede con los dos primeros métodos. GRASP-HeufJSP es una fusión de la metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP) con la heurística HeufJSP. Se presentan dos visiones distintas del azar: orientado y no orientado. También se propone el procedimiento SA-HeufJSP que une la heurística HeufJSP con la metaheurística *Simulated Annealing* (SA). Finalmente, se fusionan los dos procedimientos metaheurísticos, GRASP-HeufJSP y SA-HeufJSP, para generar otro método de resolución nombrado GRASPSA-HeufJSP. Los siguientes apartados describen los esquemas de los algoritmos propuestos, así como del procedimiento de referencia.

6.1. Método de resolución de referencia

Corominas *et al.* (2015) desarrollaron un procedimiento iterativo que aplicaba una heurística creada por los mismos autores. El proceso consiste en probar diferentes valores de un parámetro, solucionar el problema e ir guardando la mejor solución obtenida. El hecho de ir probando distintos valores del parámetro de la heurística, provoca que se esté haciendo de forma indirecta un proceso de calibrado. El procedimiento planteado puede ser considerado como enfoque integrado, es decir, los subproblemas de asignación y de programación de las operaciones se llevan a cabo a la vez.

La heurística planteada fue nombrada como HeufJSP y está constituida por dos fases que se van repitiendo en un proceso iterativo. En la primera fase, se define una secuencia de trabajos, la cual servirá para determinar el orden en que se programaran los trabajos. La segunda fase, se fundamenta en determinar una programación óptima de las operaciones según la solución en curso y sin tener en cuenta la programación de los trabajos pendientes de ser programados. Los trabajos se van programando de uno en uno y se sigue el orden definido en la primera fase. Para cada trabajo, la programación de sus operaciones se realiza construyendo un grafo de múltiples etapas, donde cada etapa representa la programación de una operación del trabajo, y encontrando su camino mínimo.

El procedimiento HeufJSP(ζ) planteado por Corominas *et al.* (2015) se fundamenta en la heurística HeufJSP, diseñada por los mismos autores. La heurística se definirá con detalle en el apartado 6.1.1. El procedimiento general está formado por dos estructuras de bucle. En la primera de ellas, el objetivo es conseguir una primera solución. En cada iteración, el parámetro ζ toma valores entre 0 y 1, con intervalos igual a Δ_{ζ}^1 , y se ejecuta HeufJSP. Esta heurística se divide en dos fases: obtener una secuencia ordenada de trabajos (Π) y encontrar, para cada trabajo, la programación óptima de sus operaciones (Y). En cada iteración, la solución obtenida se compara con la mejor encontrada hasta el momento, de manera que al final de este proceso iterativo se puede obtener la mejor solución obtenida en el primer bucle.

En el segundo bucle el proceso esencial es el mismo, para cada valor ζ se obtiene una nueva solución y se va guardando la mejor, si es que alguna mejora la solución obtenida en el primer bucle. En este caso, el parámetro ζ está dentro de un intervalo $\bar{\zeta} \pm 0,5 \cdot \Delta_{\zeta}^1$, sin tomar valores negativos ni superiores a la unidad, donde $\bar{\zeta}$ es el valor del parámetro ζ con el cual se encuentra la mejor solución en el primer bucle. Este segundo bloque iterativo se puede considerar como una optimización local alrededor de la solución encontrada en el primer bloque, que a su vez, se convierte en una calibración del parámetro ζ . El algoritmo propuesto por los autores es de tipo determinista, lo cual indica que cada vez que se ejecute

el programa se obtendrá la misma solución.

La Figura 6.1 muestra el pseudocódigo del procedimiento general de referencia para este trabajo.

```

0.  $\bar{Z} = \infty$ 
1. Para ( $\zeta = 0; \zeta \leq 1; \zeta = \zeta + \Delta_\zeta^1$ ) hacer:
    1.1.  $\Pi$  = Secuencia de trabajos inicial ( $\zeta$ )
    1.2.  $Y$  = Programación óptima ( $\Pi$ )
    1.3. Si  $Z < \bar{Z}$  entonces:
        1.3.1.  $\bar{\zeta} = \zeta$ 
        1.3.2.  $\bar{Y} = Y$ 
        1.3.3.  $\bar{Z} = Z$ 
2. Para ( $\zeta = \max(0, \bar{\zeta} - 0,5 \cdot \Delta_\zeta^1); \zeta \leq \min(1, \bar{\zeta} + 0,5 \cdot \Delta_\zeta^1); \zeta = \zeta + \Delta_\zeta^2$ ) hacer:
    2.1.  $\Pi$  = Secuencia de trabajos inicial ( $\zeta$ )
    2.2.  $Y$  = Programación óptima ( $\Pi$ )
    2.3. Si  $Z < \bar{Z}$  entonces:
        2.3.1.  $\bar{Y} = Y$ 
        2.3.2.  $\bar{Z} = Z$ 

Siendo:
 $\zeta \equiv$  parámetro de la función de ordenación de los trabajos
 $\bar{\zeta} \equiv$  parámetro  $\zeta$  con el cual se encuentra la mejor solución en el primer bucle
 $\Delta_\zeta^1, \Delta_\zeta^2 \equiv$  incremento del parámetro  $\zeta$  ( $\Delta_\zeta^2 < \Delta_\zeta^1$ )
 $Y, Z \equiv$  solución actual y valor de la función objetivo (índice de eficiencia)
 $\bar{Y}, \bar{Z} \equiv$  mejor solución y valor de la función objetivo (índice de eficiencia)
  
```

Figura 6.1: Pseudocódigo del procedimiento HeufJSP(ζ).

6.1.1. Heurística HeufJSP

Los procedimientos de resolución que se describirán en los próximos apartados hacen uso de la heurística HeufJSP en sus algoritmos. No obstante, todos los métodos hacen modificaciones de esta heurística, así como un diseño distinto de todo el procedimiento que engloba la misma.

Como se trata de una parte importante de los algoritmos propuestos, se definen con exactitud las dos fases que conforman la propia heurística HeufJSP: obtención de una secuencia de trabajos inicial y programación óptima de las operaciones de cada trabajo.

6.1.1.1. Fase 1: Obtención de una secuencia de trabajos inicial

La primera de las dos fases consiste en obtener una secuencia de trabajos, la cual se utilizará para seleccionar el orden en que se programará cada trabajo y sus respectivas operaciones en la segunda fase. Para poder ordenar los trabajos, es necesario un índice de prioridad que, en función de las propiedades, proporcionará un determinado valor a cada trabajo, el cual se empleará para establecer una secuencia en orden no decreciente.

El índice de prioridad del trabajo j , IP_j , está formado por dos conceptos: la fecha de vencimiento normalizada para el trabajo j (D_j) y la estimación normalizada de la energía necesaria para procesar el trabajo j (A_j). IP_j queda definido como la suma ponderada de ambos conceptos según el parámetro ζ ($0 \leq \zeta \leq 1$) (Ec. 6.1).

$$IP_j = \zeta \cdot D_j + (1 - \zeta) \cdot A_j \quad (\text{Ec. 6.1})$$

La normalización de un concepto consiste en dividir cada valor del conjunto por el máximo de éste. D_j se calcula dividiendo la fecha de entrega del trabajo j por la mayor fecha de entrega, considerando todos los trabajos a programar (Ec. 6.2). A_j (Ec. 6.3) es la normalización de la estimación de la energía requerida λ_j (Ec. 6.4) para procesar el trabajo j , siendo σ_{jh} una estimación de la energía necesaria para procesar la operación O_{jh} (Ec. 6.5).

$$D_j = \frac{d_j}{\max_{s=1, \dots, n} d_s} \quad (\text{Ec. 6.2})$$

$$A_j = \frac{\lambda_j}{\max_{s=1, \dots, n} \lambda_s} \quad (\text{Ec. 6.3})$$

$$\lambda_j = \sum_{h=1}^{h_j} \sigma_{jh} \quad (\text{Ec. 6.4})$$

$$\sigma_{jh} = \sum_{i \in M_{jh}} \frac{P_{ijh}}{|M_{jh}|} \quad (\text{Ec. 6.5})$$

6.1.1.2. Fase 2: Programación óptima de operaciones para cada trabajo

A partir de la secuencia de trabajos obtenida en la primera fase de la heurística, la segunda fase consiste en seleccionar el primero de los trabajos y programar sus operaciones de forma óptima. Posteriormente, se sigue el mismo procedimiento con el resto de trabajos hasta haber programado todas las operaciones de cada uno de ellos.

La disponibilidad de las máquinas es un dato conocido al principio del problema y que va variando a medida que se van programando las operaciones de los trabajos. Por lo tanto, la disponibilidad de las máquinas, cuando se programan las operaciones de un determinado trabajo, viene condicionada de las decisiones que se hayan tomado con anterioridad. Esta heurística asegura una programación de operaciones óptima para cada trabajo, pero esto no quiere decir que la solución final sea la solución óptima del problema.

Los dos subproblemas del *flexible Job-Shop Problem* se solucionan de forma integrada en esta fase. Corominas *et al.* (2015) propusieron programar las operaciones construyendo y encontrando el camino mínimo a un grafo de múltiples etapas. Tanto la etapa de construcción del grafo como la de determinación del camino mínimo se calculan de forma simultánea. En el *Anexo 1* están descritos con todo detalle los dos procesos que componen esta segunda fase y para mayor detalle se recomienda consultar Corominas *et al.* (2015).

La solución encontrada al final de la fase 2, consiste en los instantes de inicio y final de cada una de las operaciones a ser programadas, la máquina a la cual es asignada cada operación, la disponibilidad final de las máquinas y los costes asociados a la programación obtenida, que se obtiene con el índice de eficiencia compuesto.

6.2. Métodos de resolución propuestos

Se presentan siete procedimientos distintos para resolver el problema *flexible Job-Shop*. La estructura básica de la heurística HeufJSP es utilizada en todos los procedimientos, pero sufriendo modificaciones de mayor o menor importancia en cada uno de ellos.

Los procedimientos que se describen han sido nombrados como (por orden de aparición): Calibración General estática, Calibración General semidinámica, Calibración Individual estática, Calibración Individual semidinámica, GRASP-HeufJSP, SA-HeufJSP y GRASPSA-HeufJSP.

6.2.1. Calibración General estática

El primer procedimiento que se propone recibe el nombre de Calibración General estática (CGe), y consiste en solucionar el problema *flexible Job-Shop* utilizando una sola vez la heurística HeufJSP.

Se modifica, en la primera fase de la heurística, la función de ordenación o índice de prioridad de los trabajos. La función original está formada por dos atributos ponderados con parámetros que suman la unidad (Ec. 6.1). En cambio, la nueva función aumenta su complejidad añadiendo dos atributos más. Cada una de las partes está ponderada por un parámetro distinto (α^{CGe} , β^{CGe} , γ^{CGe} y ε^{CGe}) y no se impone que sumen la unidad. El índice de prioridad de este algoritmo está formado por estos cuatro atributos (Ec. 6.6): fecha de vencimiento normalizada para cada trabajo j (D_j) (Ec. 6.2), estimación normalizada de la energía necesaria para procesar el trabajo j (A_j) (Ec. 6.3), número de operaciones normalizado del trabajo j (H_j) y promedio normalizado del tiempo de proceso del trabajo j (T_j).

$$IP_j^{CGe} = \alpha^{CGe} \cdot D_j + \beta^{CGe} \cdot A_j + \gamma^{CGe} \cdot H_j + \varepsilon^{CGe} \cdot T_j \quad (\text{Ec. 6.6})$$

Siendo las fórmulas para calcular los nuevos conceptos:

$$H_j = \frac{h_j}{\max_{s=1, \dots, n} h_s} \quad (\text{Ec. 6.7})$$

$$T_j = \frac{\tau_j}{\max_{s=1, \dots, n} \tau_s} \quad (\text{Ec. 6.8})$$

Donde τ_j es el promedio del tiempo de proceso del trabajo j :

$$\tau_j = \sum_{h=1}^{h_j} \left(\frac{\sum_{i \in M_{jh}} p_{ijh}}{|M_{jh}|} \right) \quad (\text{Ec. 6.9})$$

Con este procedimiento se busca encontrar programaciones con mayor rapidez, ya que sólo se aplica una vez la heurística HeufJSP, a diferencia del procedimiento de referencia que ejecuta la heurística HeufJSP con distintos valores de ζ . El pseudocódigo del algoritmo es el siguiente (Figura 6.2):

0. Definición de los valores que toman los parámetros del índice de prioridad IP_j^{CGe} .
1. Obtención de la secuencia de ordenación de los trabajos. Se utiliza el índice de prioridad IP_j^{CGe} para asociar un valor a cada trabajo y así poderlos ordenar según un orden no decreciente.
2. Programación de las operaciones de los trabajos, siguiendo el orden definido en el paso 1. La programación de cada trabajo se consigue construyendo y calculando el camino mínimo de un grafo de múltiples etapas.
3. Cálculo del valor del índice de eficiencia compuesto.

Figura 6.2: Pseudocódigo del procedimiento Calibración General estática.

Se trata de un procedimiento determinista. Esto significa que cada vez que se ejecute este procedimiento para un mismo ejemplar del problema, se obtendrá la misma solución.

La palabra estática del nombre del algoritmo proviene de la naturaleza de la función de ordenación o índice de prioridad. Para cada trabajo j , los valores que toman los conceptos que forman el índice de prioridad siempre son los mismos, independientemente del momento de la programación en la cual se esté. Por lo tanto, solo es necesario ordenar los trabajos antes de empezar la programación de los mismos.

En el pseudocódigo del algoritmo se ha definido un paso 0 que es la definición de los parámetros. Para encontrar los valores más adecuados, se realiza un proceso de calibración previo a la ejecución del algoritmo. Los términos “calibración general” del nombre del procedimiento hacen referencia a que los valores de los parámetros calibrados son los mismos para todos los ejemplares.

El proceso de calibración se realiza aplicando la metodología *Empirically Adjusted Greedy Heuristics* (EAGH), definida por Corominas y Pastor (2011). El índice de prioridad puede verse como una función f de varios parámetros. Esto conlleva que exista un conjunto F de infinitos índices de prioridad, fruto del número infinito de combinaciones de valores que pueden tomar los parámetros (valores reales). EAGH es un procedimiento que consiste en encontrar la mejor función $f \in F$ para un conjunto de ejemplares de entrenamiento o calibración. Para ello, se aplica un algoritmo directo de optimización no lineal a la función φ , la cual se define como la suma de los valores del índice de eficiencia al solucionar los ejemplares de calibración con la función de ordenación f . El algoritmo directo que se utiliza es *Nelder & Mead*, diseñado por Nelder y Mead (1965). En el Anexo 2, se puede encontrar una descripción más amplia del método EAGH, así como del algoritmo de optimización directo utilizado, *Nelder & Mead*, y para mayor detalle se recomienda consultar Corominas y Pastor (2011).

El pseudocódigo que se sigue en el proceso de calibración es presentado en la Figura 6.3:

0. Definición del punto inicial y de la longitud de las aristas del hipertetraedro regular inicial (L) necesarios para inicializar el algoritmo *Nelder & Mead*. El punto inicial tiene tantas coordenadas como parámetros, en este caso son 4. Definición de las dos condiciones de fin.
1. Inicio del algoritmo *Nelder & Mead*. Hasta cumplir con las dos condiciones de fin:
 - 1.1. Generar en cada iteración un punto o valores de los parámetros, los cuales ponderan los atributos del índice de prioridad.
 - 1.2. Para todos los ejemplares de calibrado:
 - 1.2.1. Aplicar las dos fases de la heurística HeufJSP: obtener un secuencia de los trabajos, utilizando los parámetros generados por el algoritmo *Nelder & Mead* en el paso 1.1; y la programación de los trabajos, según el orden de los trabajos.
 - 1.2.2. Calcular el valor del índice de eficiencia compuesto para la solución obtenida.
 - 1.2.3. Suma acumulada de los índices de eficiencia de los ejemplares. Al terminar el paso 1.2, se habrá obtenido el valor de la función φ , el cual *Nelder & Mead* intenta minimizar para el método EAGH.
 - 1.3. Comprobación del cumplimiento de las condiciones de fin. Si no se cumplen las dos condiciones, volver al paso 1.1. En caso contrario, FIN del proceso de calibración de los parámetros.

Figura 6.3: Pseudocódigo del proceso de calibración del procedimiento Calibración General estática.

6.2.2. Calibración General semidinámica

Una desventaja que presenta el proceso de Calibración General estática es que la ordenación de los trabajos sólo se hace teniendo en cuenta la disponibilidad inicial de las máquinas. Para solventar esta deficiencia, se presenta el segundo procedimiento de este trabajo: Calibración General semidinámica (CGs). El valor del índice de prioridad o función de ordenación puede cambiar durante la construcción de la solución, según la disponibilidad de las máquinas en la solución parcial. Por este motivo, se reordena la secuencia de trabajos pendientes de ser programados tres veces: al inicio del procedimiento, una vez programados el 33% de los trabajos y programados el 66% de los trabajos. De aquí la denominación de semidinámica, ya que si se ordenara después de programar cada trabajo, se consideraría como dinámica. El caso opuesto a la dinámica sería ordenarlo sólo una vez (Calibración General estática).

El motivo por el cual se ha fijado esta proporción y no otra radica en la relación: calidad de la

solución encontrada - recursos de tiempo de CPU necesarios. Con un breve experimento computacional, se ha comprobado que reordenando los trabajos más de tres veces, la mejora de la calidad es muy pequeña y, en cambio, el tiempo necesario para resolver los ejemplares aumenta mucho. Cabe añadir, además, que las variaciones en el orden de los trabajos se ven patentes cuando ya se han programado bastantes trabajos, con lo que reordenar cada pocos trabajos es ineficaz.

El índice de prioridad o función de ordenación semidinámica, IP_j^{CGs} , está formado por cinco atributos, cada uno ponderado por su respectivo parámetro a calibrar. Tres de los conceptos son compartidos con el índice de prioridad estático (IP_j^{CGe}): fecha de vencimiento normalizada para cada trabajo j (D_j) (Ec. 6.2), estimación normalizada de la energía necesaria para procesar el trabajo j (A_j) (Ec. 6.3) y número de operaciones normalizado del trabajo j (H_j) (Ec. 6.7). Estos atributos son invariables a lo largo del proceso de calibración y son los dos restantes los que proporcionan dinamismo a la función de ordenación, permitiendo la posibilidad que aparezcan variaciones en la ordenación de los trabajos. Los dos conceptos nuevos son:

- Estimación normalizada del coste de la energía necesaria para producir el trabajo j (EE_j). Corresponde al sumatorio de una estimación del coste de la energía necesaria para producir cada operación del trabajo j . La estimación del coste de la energía necesaria para cada operación, se obtiene con el sumatorio de la energía necesaria en cada una de las máquinas que podrían procesarla, multiplicada por el coste medio de la energía en esta máquina hasta la fecha de vencimiento del trabajo j , teniendo en cuenta sólo los períodos en que la operación podría ser procesada de forma completa. El sumatorio se divide entre el número de máquinas que pueden llegar a ejecutar la operación. En el cálculo del coste medio de la energía se tiene en cuenta la discriminación horaria, que distingue entre periodos con tarifa valle y periodos con tarifa punta.
- Estimación normalizada de la duración del tiempo de proceso del trabajo j (ET_j). Corresponde al sumatorio de la estimación de la duración del tiempo de proceso de cada una de las operaciones del trabajo j . La estimación del tiempo de proceso de cada operación es el sumatorio del tiempo de proceso necesario en cada una de las máquinas que podrían procesarla, teniendo en cuenta sólo los períodos en que la operación podría ser procesada de forma completa, dividido entre el número de máquinas que podrían procesarla.

La expresión del índice de prioridad semidinámico queda definida como (Ec. 6.10):

$$IP_j^{CGs} = \alpha^{CGs} \cdot D_j + \beta^{CGs} \cdot A_j + \gamma^{CGs} \cdot H_j + \varepsilon^{CGs} \cdot EE_j + \sigma^{CGs} \cdot ET_j \quad (\text{Ec. 6.10})$$

Los pseudocódigos presentados en la Calibración General estática sufren modificaciones para la Calibración General semidinámica, ya que se reordena la secuencia de los trabajos. La enumeración de los pasos para la calibración de los parámetros, que en este caso son cinco, son los siguientes (Figura 6.4):

0. Definición del punto inicial y de la longitud de las aristas del hipertetraedro regular inicial (L) necesarios para inicializar el algoritmo *Nelder & Mead*. El punto inicial tiene tantas coordenadas como parámetros, en este caso son 5. Definición de las dos condiciones de fin.
1. Inicio del algoritmo *Nelder & Mead*. Hasta cumplir con las dos condiciones de fin:
 - 1.1. Generar en cada iteración un punto o valores de los parámetros, los cuales ponderan los atributos del índice de prioridad.
 - 1.2. Para todos los ejemplares de calibrado:
 - 1.2.1. Iniciar contador ($contador = 0$):
 - 1.2.1.1. Ordenar los trabajos no programados para encontrar una secuencia utilizando el índice de eficiencia IP_j^{CGs} .
 - 1.2.1.2. Seleccionar el primer trabajo de la secuencia ordenada y programar sus operaciones con el método del grafo de múltiples etapas. Incrementar en uno el *contador* ($contador = contador + 1$).
 - 1.2.1.3. Eliminar el trabajo programado de la secuencia ordenada.
 - 1.2.1.4. Si no quedan trabajos para programar, ir al paso 1.2.2. En caso contrario, si $contador = round((1/3) \cdot n)$, ir al paso 1.2.1.1 y reinicializar el contador ($contador = 0$). Sino, volver al paso 1.2.1.2.
 - 1.2.2. Calcular el valor del índice de eficiencia compuesto para la solución obtenida.
 - 1.2.3. Suma acumulada de los índices de eficiencia de los ejemplares. Al terminar el paso 1.2, se habrá obtenido el valor de la función φ , la cual *Nelder & Mead* intenta minimizar para el método EAGH.
 - 1.3. Comprobación del cumplimiento de las condiciones de fin. Si no se cumplen las dos condiciones, volver al paso 1.1. En caso contrario, FIN del proceso de calibración de los parámetros.

Figura 6.4: Pseudocódigo proceso de calibración del procedimiento Calibración General semidinámica.

De la misma manera, el pseudocódigo del procedimiento de resolución de un ejemplar (Figura 6.5), por la aplicación de Calibración General semidinámica, sufre cambios:

0. Definición de los valores que toman los parámetros del índice de prioridad IP_j^{CGs} .
1. Heurística HeufJSP con ordenación semidinámica. Iniciar contador ($contador = 0$):
 - 1.1. Ordenar los trabajos no programados para encontrar una secuencia.
 - 1.2. Seleccionar el primer trabajo de la secuencia ordenada y programar sus operaciones con el método del grafo de múltiples etapas. Incrementar en uno el *contador*.
 - 1.3. Eliminar el trabajo programado de la secuencia ordenada.
 - 1.4. Si no quedan trabajos para programar, ir al paso 2. En caso contrario, si $contador = (1/3) \cdot n$, ir al paso 1.1 y reinicializar el contador ($contador = 0$). Sino, volver al paso 1.2.
2. Cálculo del valor del índice de eficiencia compuesto.

Figura 6.5: Pseudocódigo del procedimiento Calibración General semidinámica.

6.2.3. Calibración Individual estática

En los dos algoritmos de Calibración General se buscan unos parámetros que serán comunes para resolver cualquier ejemplar, sean cuales sean sus características. El procedimiento de Calibración Individual estática (Cle) tiene el objetivo de adaptar los parámetros que ponderan los atributos del índice de prioridad para cada ejemplar en particular. Por lo tanto, dos ejemplares distintos pueden obtener valores de los parámetros diferentes al aplicar el planteamiento resolutivo presentado en este apartado.

La idea fundamental de Calibración Individual estática se puede llegar a comparar con el método definido por Corominas *et al.* (2015). Su procedimiento se basaba en mejorar la solución obtenida variando el parámetro ζ , lo cual se puede equiparar a una especie de calibración específica del parámetro ζ para cada ejemplar.

A diferencia de los algoritmos de Calibración General, el proceso de calibración no se realiza con el método *Empirically Adjusted Greedy Heuristics* (EAGH), sino con el método EAGH-1 presentado por Corominas *et al.* (2014). Esto es gracias al hecho que se adapta el índice de prioridad a cada ejemplar en particular y no se busca un único índice de prioridad para todos los ejemplares, por lo tanto, únicamente es necesaria la ejecución de EAGH para un ejemplar. El algoritmo de optimización directo que se utiliza sigue siendo *Nelder & Mead*.

El pseudocódigo que sigue el procedimiento de Calibración Individual estática se presenta a continuación (Figura 6.6). Hay que tener en cuenta que la última iteración del algoritmo de *Nelder & Mead* es la solución del problema, donde se obtiene la programación de las operaciones, el valor del índice de eficiencia compuesto y los parámetros calibrados de la

función de ordenación o índice de prioridad.

0. Definición del punto inicial y de la longitud de las aristas del hipertetraedro regular inicial (L), necesarios para inicializar el algoritmo *Nelder & Mead*. El punto inicial tiene tantas coordenadas como parámetros. Definición de las dos condiciones de fin.
1. Inicio del algoritmo *Nelder & Mead*. Hasta cumplir con las dos condiciones de fin:
 - 1.1. Generar en cada iteración un punto o valores de los parámetros, los cuales ponderan los atributos del índice de prioridad.
 - 1.2. Aplicar las dos fases de la heurística HeufJSP: obtener una secuencia de los trabajos, utilizando los parámetros generados por el algoritmo *Nelder & Mead* en el paso 1.1; y la programación de los trabajos, según el orden de los trabajos.
 - 1.3. Calcular el valor del índice de eficiencia compuesto para la solución obtenida.
 - 1.4. Comprobación del cumplimiento de las condiciones de fin. Si no se cumplen las dos condiciones, volver al paso 1.1. En caso contrario, FIN del procedimiento de Calibración Individual estática.

Figura 6.6: Pseudocódigo del procedimiento de Calibración Individual estática.

En el procedimiento de Calibración General estática se ha diseñado un índice de prioridad compuesto por cuatro atributos: fecha de vencimiento normalizada para cada trabajo j (D_j), estimación normalizada de la energía necesaria para procesar el trabajo j (A_j), número de operaciones normalizado del trabajo j (H_j) y el promedio normalizado del tiempo de proceso del trabajo j (T_j). Para el procedimiento que se presenta en este apartado, se decide hacer tres combinaciones con los atributos H_j y T_j . En todas las combinaciones, los dos primeros atributos, D_j y A_j , son presentes.

Fruto de la combinación de los atributos, surgen tres índices de prioridad distintos. Cada una de estas funciones de ordenación se evaluará de forma separada para comprobar su calidad. Las expresiones de los índices de prioridad son:

$$IP_j^{Cle_1} = \alpha^{Cle_1} \cdot D_j + \beta^{Cle_1} \cdot A_j + \gamma^{Cle_1} \cdot H_j \quad (\text{Ec. 6.11})$$

$$IP_j^{Cle_2} = \alpha^{Cle_2} \cdot D_j + \beta^{Cle_2} \cdot A_j + \varepsilon^{Cle_2} \cdot T_j \quad (\text{Ec. 6.12})$$

$$IP_j^{Cle_3} = \alpha^{Cle_3} \cdot D_j + \beta^{Cle_3} \cdot A_j + \gamma^{Cle_3} \cdot H_j + \varepsilon^{Cle_3} \cdot T_j \quad (\text{Ec. 6.13})$$

6.2.4. Calibración Individual semidinámica

Similar a lo que sucede entre los dos procedimientos de Calibración General, el algoritmo planteado como Calibración Individual estática sólo ordena los trabajos una vez, antes de iniciar la programación de las operaciones. El motivo reside en que los cuatro atributos son estáticos, es decir, sus valores son permanentemente los mismos a lo largo de la programación.

Se decide implementar el índice de prioridad semidinámico definido en el procedimiento de Calibración General semidinámica, en una estructura de Calibración Individual. El objetivo es evaluar el comportamiento del algoritmo teniendo en cuenta las decisiones de programación tomadas y la variación de la disponibilidad de las máquinas. Se mantiene el número de ordenaciones en tres y los porcentajes: una al principio y dos durante el proceso de resolución del problema (una vez programados el 33% y el 66% de los trabajos).

El índice de prioridad semidinámico individual, común con la Calibración General semidinámica, está formado por cinco atributos (Ec. 6.10): fecha de vencimiento normalizada para cada trabajo j (D_j), estimación normalizada de la energía necesaria para procesar el trabajo j (A_j), número de operaciones normalizado del trabajo j (H_j), estimación normalizada del coste de la energía necesaria para producir el trabajo j (EE_j) y estimación normalizada de la duración del tiempo de proceso del trabajo j (ET_j). Los dos últimos son los que dan dinamismo a la función de ordenación de los trabajos.

El pseudocódigo del procedimiento Calibración Individual semidinámica (CIs) sufre modificaciones con respecto a la estática. Los pasos a seguir son (Figura 6.7):

0. Definición del punto inicial y de la longitud de las aristas del hipertetraedro regular inicial (L) necesarios para inicializar el algoritmo *Nelder & Mead*. El punto inicial tiene tantas coordenadas como parámetros, en este caso son 5. Definición de las dos condiciones de fin.
1. Inicio del algoritmo *Nelder & Mead*. Hasta cumplir con las dos condiciones de fin:
 - 1.1. Generar en cada iteración un punto o valores de los parámetros, los cuales ponderan los atributos del índice de prioridad.
 - 1.2. Iniciar contador ($contador = 0$):
 - 1.2.1. Ordenar los trabajos no programados para encontrar una secuencia.
 - 1.2.2. Seleccionar el primer trabajo de la secuencia ordenada y programar sus operaciones con el método del grafo de múltiples etapas. Incrementar en uno el *contador*.
 - 1.2.3. Eliminar el trabajo programado de la secuencia ordenada.

- 1.2.4. Si no quedan trabajos para programar, ir al paso 1.3. En caso contrario, si $contador = round((1/3) \cdot n)$, ir al paso 1.2.1 y reinicializar el contador ($contador = 0$). Sino, volver al paso 1.2.2.
- 1.3. Calcular el valor del índice de eficiencia compuesto para la solución obtenida.
- 1.4. Comprobación del cumplimiento de las condiciones de fin. Si no se cumplen las dos condiciones, volver al paso 1.1. En caso contrario, FIN del procedimiento de Calibración Individual semidinámica.

Figura 6.7: Pseudocódigo del procedimiento de Calibración Individual semidinámica.

6.2.5. GRASP-HeufJSP

Hasta ahora se han presentado heurísticas deterministas, es decir, aquellas que si se ejecutan con las mismas condiciones iniciales y sobre el mismo ejemplar del problema de estudio, se obtendría la misma solución. A partir de este apartado, se proponen procedimientos de resolución del problema *flexible Job-Shop* mediante el uso de procesos estocásticos. Literalmente, un proceso estocástico se puede definir como aquel que no es determinista y, por lo tanto, la solución obtenida es fruto tanto de la combinación de las acciones predecibles del proceso como por la intervención de elementos aleatorios.

Las metaheurísticas son estrategias iterativas que combinan diferentes conceptos para explorar el espacio de soluciones, permitiendo escapar de óptimos locales para intentar llegar hacia el óptimo global. Se ha mencionado la palabra intentar ya que, según las características del problema y del tipo de metaheurística empleada, los recursos necesarios para lograr el óptimo global pueden llegar a ser muy grandes. La condición de fin de las metaheurísticas es en general arbitraria y, normalmente, suele ser por tiempo o por número de iteraciones de la propia metaheurística, entre otras opciones.

La metaheurística en la que se basa el procedimiento propuesto en este apartado recibe el nombre *Greedy Randomized Adaptive Search Procedure*, a partir de ahora GRASP (Feo y Resende 1989). Ésta consiste en encontrar s soluciones, aplicarles un proceso de mejora, normalmente optimización local, y quedarse la mejor. La generación de soluciones iniciales se realiza con una heurística con azar orientado, en general una heurística *greedy* o voraz. Una heurística *greedy* consiste en ordenar una lista de objetos según un criterio predefinido y, hasta que no queden objetos, seleccionar el primer objeto para ser utilizado en un proceso concreto del problema a resolver. Para la metaheurística GRASP, en cada selección de la heurística *greedy* se determina una lista de objetos candidatos, de la cual no se selecciona el candidato con mejor índice, sino uno al azar perteneciente a una lista restringida de los k mejores. La probabilidad de elección es proporcional a su valor

asociado. Este método de selección se repite hasta que la lista de candidatos esté vacía. El pseudocódigo de la metaheurística GRASP teórica, con una condición de fin por número de iteraciones, es la siguiente (Figura 6.8):

0. $s = 1$ y $\bar{Z} = \infty$

1. Para ($s = 1; s \leq S; s = s + 1$) hacer:

1.1. Obtener una solución inicial con azar orientado (X) y el valor de su índice de eficiencia (Z).

1.2. Aplicar un proceso de optimización local a la solución X para obtener X' .

1.3. Si $Z' < \bar{Z}$, X' pasa a ser la mejor solución ($\bar{X} = X'$ y $\bar{Z} = Z'$).

Siendo:

s \equiv Número de iteración actual.

S \equiv Condición de fin de la metaheurística por número de iteraciones.

\bar{X} \equiv Mejor solución encontrada hasta el momento.

\bar{Z} \equiv Valor del índice de eficiencia de la mejor solución encontrada hasta el momento.

X \equiv Solución encontrada por azar orientado

Z \equiv Valor del índice de eficiencia de la solución encontrada por azar orientado.

X' \equiv Solución vecina de X después de aplicarle optimización local.

Z' \equiv Valor del índice de eficiencia de X' después de aplicarle optimización local.

Figura 6.8: Pseudocódigo de la metaheurística GRASP teórica.

El procedimiento GRASP-HeufJSP planteado en este apartado presenta variaciones respecto al procedimiento teórico de la metaheurística GRASP y de la heurística HeufJSP. Estas variaciones se han realizado por motivos de consumo de recursos de tiempo. Se ha comprobado que los ejemplares del problema, que se pueden categorizan como medianos o grandes, necesitan de un tiempo considerable para ser resueltos, lo que provoca que sea inviable aplicar optimización local a cada una de las soluciones obtenidas. Por lo tanto, se decide hacer un primer proceso de búsqueda de soluciones con azar orientado durante un tiempo limitado, se selecciona la mejor solución obtenida y, únicamente a ésta, se le aplica un proceso de optimización local, también limitado por tiempo.

La metaheurística GRASP tiene el tamaño de la lista restringida como un parámetro importante a calibrar. Para ello se aplicará el procedimiento llamado CALIBRA (Adenso-Díaz y Laguna 2006), creado para el calibrado de parámetros. Está basado en usar conjuntamente el diseño de experimentos factorial de Taguchi y un proceso de búsqueda local. En el Anexo 2 se puede encontrar más información sobre el software CALIBRA.

No se calibrará directamente el tamaño de la lista restringida (k), sino un parámetro

$k' \in [0.1, 1]$ que multiplica al número total de trabajos a programar, siendo el redondeo al entero más cercano el tamaño de la lista. Como mínimo se fija que la lista tenga un tamaño de 3 trabajos, siempre y cuando aún queden trabajos suficientes para programar, para así poder asegurar en los ejemplares con pocos trabajos que el algoritmo tiene un tamaño aceptable de la lista restringida con el cual aplicar el azar.

$$k = \min(n^{\circ} \text{trabajos no programados}, \max(k' \cdot n, 3)) \quad (\text{Ec. 6.14})$$

A continuación, se muestra un ejemplo de aplicación de la ecuación (Ec. 6.14) para calcular la longitud de la lista restringida en un problema con 10 trabajos a programar (Tabla 6.1):

	Trabajos programados									
	0	1	2	3	4	5	6	7	8	9
$n = 10$ $k' = 0.1$	3	3	3	3	3	3	3	3	2	1
$n = 10$ $k' = 0.5$	5	5	5	5	5	5	4	3	2	1

Tabla 6.1: Ejemplo del cálculo del tamaño de la lista restringida.

En los procedimientos descritos en apartados anteriores, la secuencia de trabajos se clasificaba en orden no decreciente según el valor del índice de prioridad que tomaba cada trabajo. Esta secuencia se utilizaba para determinar qué trabajo era el siguiente a programar, seleccionando el situado en primera posición en la secuencia. Para esta versión modificada de la metaheurística GRASP, la secuencia ordenada de los trabajos será la lista de candidatos, a partir de la cual se seleccionarán los k trabajos con mejor índice de prioridad, que compondrán la lista restringida. El índice de prioridad que se utiliza es el mismo que en el procedimiento de Calibración General estática, con el mismo valor de los parámetros ($IP^{GRASP} = IP^{CGe}$).

Los trabajos de la lista restringida tendrán asignado un rango de aceptación según el valor del índice de prioridad. La selección del siguiente trabajo a programar se efectúa generando un valor al azar y comprobando en que rango se encuentra. Se distinguen dos métodos para calcular el rango de aceptación: con azar orientado (clásico) y con azar no orientado (innovación):

- Con azar orientado: Pertenece al procedimiento GRASP original y se caracteriza por dar mayor probabilidad a los candidatos mejor clasificados en la lista restringida (Tabla 6.2 con $k = 4$).

Lista candidatos		Lista restringida					
Trabajo	IP^{GRASP}	Trabajo	IP^{GRASP}	$\frac{1}{IP^{GRASP}}$	$\Delta = \frac{1 / IP^{GRASP}}{\sum (1 / IP^{GRASP})}$	$\Delta_{acu.}$	Rango
9	1.283	9	1.283	0.779	0.274	0.274	$[0, 0.274)$
4	1.321	4	1.321	0.757	0.266	0.540	$[0.274, 0.54)$
1	1.358	1	1.358	0.736	0.259	0.799	$[0.54, 0.799)$
7	1.747	7	1.747	0.572	0.201	1	$[0.799, 1)$
2	2.511			2.844			
5	2.580						
6	2.904						
8	3.138						

Tabla 6.2: Ejemplo del cálculo de los rangos de probabilidad para GRASP-HeufJSP con azar orientado.

- Con azar no orientado: Se trata de un nuevo enfoque de la orientación del azar en la metaheurística GRASP. Se da más probabilidad a los trabajos que están peor clasificados en la lista restringida (Tabla 6.3 con $k = 4$).

Lista candidatos		Lista restringida				
Trabajo	IP^{GRASP}	Trabajo	IP^{GRASP}	$\Delta = \frac{IP^{GRASP}}{\sum (IP^{GRASP})}$	$\Delta_{acu.}$	Rango
9	1.283	9	1.283	0.225	0.225	$[0, 0.225)$
4	1.321	4	1.321	0.231	0.456	$[0.225, 0.456)$
1	1.358	1	1.358	0.238	0.694	$[0.456, 0.694)$
7	1.747	7	1.747	0.306	1	$[0.694, 1)$
2	2.511			5.709		
5	2.580					
6	2.904					
8	3.138					

Tabla 6.3: Ejemplo del cálculo de los rangos de probabilidad para GRASP-HeufJSP con azar no orientado.

Los algoritmos de optimización local tienen el objetivo de encontrar un óptimo local, es decir, una solución tal que en su entorno (vecindario) no exista una solución mejor que ella misma. Según la estrategia de exploración en el vecindario, estos procesos se pueden clasificar en dos tipos:

- Algoritmo Exhaustivo de Descenso (AED): Consiste en generar todas las soluciones vecinas de la solución inicial (Z) y seleccionar la mejor vecina (Z'). Si, en caso de

minimización, $Z' < Z$, Z' sustituye a Z ($Z = Z'$) y se repite el proceso. En caso contrario, se termina el algoritmo y Z es el óptimo local.

- **Algoritmo No Exhaustivo de Descenso (ANED):** Consiste en generar una solución vecina (Z') de la solución inicial (Z). Si, en caso de minimización, $Z' < Z$, Z' sustituye a Z y se genera otra vecina de $Z = Z'$. En caso contrario, no se sustituye Z y se genera otra vecina. En el caso que Z' sea la última vecina de Z , finaliza el algoritmo de optimización y Z es el óptimo local.

La metaheurística GRASP que se propone utiliza un Algoritmo No Exhaustivo de Descenso (ANED) para la etapa de optimización local. Además, la condición de fin será llegar a un óptimo local o superar un límite de tiempo. El método escogido para generar el vecindario consiste en intercambiar la posición en la secuencia de dos trabajos, consecutivos o no consecutivos. Se parte de la secuencia de trabajos ordenada de la mejor solución encontrada hasta el momento. Consiste en generar dos números enteros al azar no mayores al número total de trabajos. Se intercambian los trabajos de las posiciones que indican los dos números generados al azar. Finalmente, se programan las operaciones de los trabajos, teniendo en cuenta el orden de la nueva secuencia generada y aplicando el método del grafo de múltiples etapas. La Figura 6.9 ejemplifica la generación de una secuencia de trabajos vecina.

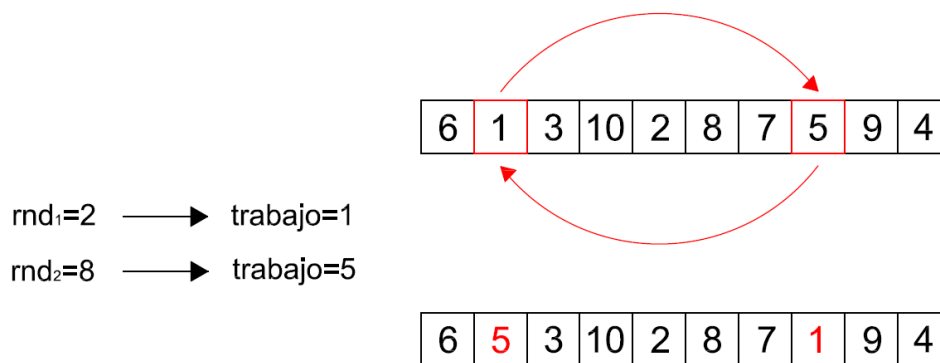


Figura 6.9: Ejemplo que representa el proceso para encontrar una secuencia vecina en GRASP-HeufJSP.

El número de vecinas (combinaciones) se calcula con la expresión $n \cdot (n-1) / 2$. Acto seguido, se muestra un ejemplo para calcular el tamaño del vecindario (Ec. 6.15).

$$\frac{n \cdot (n-1)}{2} = \frac{5 \cdot (5-1)}{2} = 10 \text{ vecinas} \quad (\text{Ec. 6.15})$$

El pseudocódigo del procedimiento GRASP-HeufJSP basado en la metaheurística GRASP teórica, definida y adaptada en este apartado, es el siguiente (Figura 6.10):

0. Iniciar a cero el temporizador t , valor del índice de eficiencia compuesto de la mejor solución $\bar{Z} = \infty$ y de la mejor solución por azar $\bar{Z}^{azar} = \infty$. Definición de las condiciones de fin: t^{azar} y t^{opt} .
1. Mientras $t < t^{azar}$:
 - 1.1. Obtención de una solución por azar orientado o desorientado (en función de la versión deseada).
 - 1.1.1. Ordenar los trabajos con el índice de prioridad IP^{GRASP} para encontrar una secuencia, la cual será la lista de candidatos. Hacer hasta que la lista de candidatos esté vacía:
 - 1.1.1.1. Cálculo del tamaño de la lista restringida (k), así como los rangos de aceptación.
 - 1.1.1.2. Generar un número al azar ($rnd \in [0,1)$) y escoger el trabajo que contenga el valor.
 - 1.1.1.3. Programar de forma óptima las operaciones del trabajo seleccionado con el método del grafo de múltiples etapas. Eliminar el trabajo ya programado de la lista de candidatos.
 - 1.1.2. Calcular el valor del índice de eficiencia compuesto de la solución X^{azar} (Z^{azar}).
 - 1.2. Si $Z^{azar} < \bar{Z}^{azar}$, entonces la mejor solución encontrada por azar pasa a ser X^{azar} ($\bar{X}^{azar} = X^{azar}$ y $\bar{Z}^{azar} = Z^{azar}$).
2. La mejor solución encontrada hasta el momento es la mejor solución encontrada por azar: $\bar{X} = \bar{X}^{azar}$ y $\bar{Z} = \bar{Z}^{azar}$. Mientras $t < (t^{azar} + t^{opt})$ o queden vecinas por visitar:
 - 2.1. Generar la solución vecina X' de \bar{X} y calcular el valor de su índice de eficiencia compuesto.
 - 2.2. Si X' es mejor que \bar{X} ($Z' < \bar{Z}$), entonces X' sustituye a \bar{X} ($\bar{Z} = Z'$) e ir al paso 2.1. En caso contrario, si X' es la última vecina de \bar{X} , FIN del proceso de optimización y del procedimiento GRASP-HeufJSP; sino ir al paso 2.1.

Siendo:

$\bar{X} \equiv$ Mejor solución encontrada hasta el momento.

$\bar{X}^{azar} \equiv$ Mejor solución encontrada por azar.

$X^{azar} \equiv$ Solución encontrada por azar.

$X' \equiv$ Solución vecina de la mejor solución hallada hasta el momento \bar{X} .

$\bar{Z} \equiv$ Valor del índice de eficiencia compuesto de la mejor solución hallada \bar{X} .

$\bar{Z}^{azar} \equiv$ Valor del índice de eficiencia compuesto de la mejor solución por azar \bar{X}^{azar} .

$Z^{azar} \equiv$ Valor del índice de eficiencia compuesto de la solución por azar X^{azar} .
 $Z' \equiv$ Valor del índice de eficiencia compuesto de la solución vecina X' .

Figura 6.10: Pseudocódigo del procedimiento GRASP-HeufJSP con azar orientado o no orientado.

6.2.6. SA-HeufJSP

La segunda metaheurística que se presenta en este trabajo para resolver el problema *flexible Job-Shop* se llama *Simulated Annealing* (SA) o Recocido Simulado (Kirkpatrick *et al.* 1983). La combinación de esta metaheurística con la heurística HeufJSP conforma el procedimiento diseñado con el nombre SA-HeufJSP.

La ejecución de la metaheurística GRASP realiza un primer proceso en que se hace una amplia exploración del espacio de soluciones y a la mejor solución encontrada se le aplica un algoritmo de optimización local (Figura 6.11a). En cambio, *Simulated Annealing* parte de una solución inicial y tiende hacia un óptimo local (Figura 6.11b), pudiendo escapar de él y dirigirse a otros óptimos locales. Por el camino, no sólo acepta soluciones vecinas mejores, sino con cierta probabilidad también de peores, lo que le permite escapar de óptimos locales. Las siguientes imágenes muestran una comparativa gráfica del funcionamiento de las dos metaheurísticas:

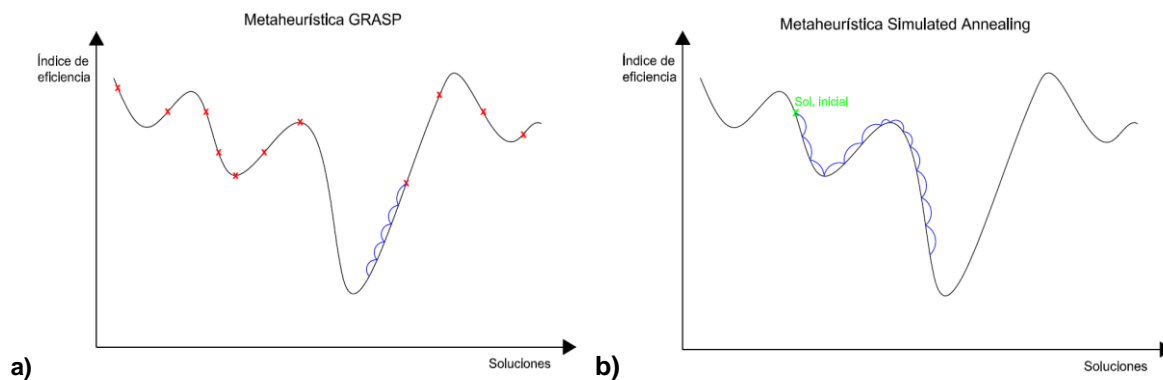


Figura 6.11: a) Gráfica que ejemplifica el comportamiento del procedimiento GRASP-HeufJSP y b) gráfica que ejemplifica el comportamiento del procedimiento SA-HeufJSP.

La metaheurística *Simulated Annealing* planteada en este apartado genera la solución inicial mediante la ejecución del algoritmo Calibración General estática. Se utiliza el mismo índice de prioridad para generar la secuencia de trabajos, la cual determina el orden en que se van seleccionando los trabajos para ser programados. También, se utilizan los mismos valores de los parámetros encontrados en el proceso de calibración del procedimiento de Calibración General estática. El índice de prioridad utilizado en SA-HeufJSP queda definido

en la ecuación (Ec. 6.16).

$$IP_j^{SA} = IP_j^{CGe} = \alpha^{CGe} \cdot D_j + \beta^{CGe} \cdot A_j + \gamma^{CGe} \cdot H_j + \varepsilon^{CGe} \cdot T_j \quad (\text{Ec. 6.16})$$

A continuación, se enumeran los pasos que conforman el pseudocódigo del procedimiento *Simulated Annealing* (Figura 6.12):

0. Obtener una solución inicial:
 - 0.1. Calcular la secuencia de ordenación de los trabajos con el índice de prioridad IP^{SA} .
 - 0.2. Programación óptima de los trabajos (X^{cur}), siguiendo el orden definido en el paso 0.1. La programación de cada trabajo se realiza construyendo y calculando el camino mínimo de un grafo de múltiples etapas.
 - 0.3. Cálculo del valor del índice de eficiencia compuesto Z^{cur} de la solución inicial X^{cur} .
 - 0.4. Definir la solución inicial como la mejor solución encontrada hasta el momento: $\bar{X} = X^{cur}$ y $\bar{Z} = Z^{cur}$.
1. Hasta la superación de un tiempo predeterminado (condición de fin):
 - 1.1. Generar una solución vecina, X^{vec} , a partir de la solución actual X^{cur} . Calcular el valor del índice de eficiencia compuesto de la solución vecina Z^{vec} .
 - 1.2. Proceso de selección de la solución vecina. Si Z^{vec} no es peor que Z^{cur} ($Z^{vec} \leq Z^{cur}$), entonces la solución vecina pasa a ser la solución actual: $X^{cur} = X^{vec}$ y $Z^{cur} = Z^{vec}$. Si además la solución vecina es mejor que la mejor solución encontrada hasta el momento: $\bar{X} = X^{vec}$ y $\bar{Z} = Z^{vec}$. En caso contrario, se genera un número al azar $rnd \in [0,1]$ y si $rnd < \prod(Z^{cur}, Z^{vec})$, entonces se acepta la solución que es peor que la actual: $X^{cur} = X^{vec}$ y $Z^{cur} = Z^{vec}$. Comprobar la condición de fin, sino se cumple volver al paso 1.1.

Siendo:

$\bar{X} \equiv$ Mejor solución encontrada hasta el momento.

$X^{cur} \equiv$ Solución en curso o actual.

$X^{vec} \equiv$ Solución vecina de la solución en curso X^{cur} .

$\bar{Z} \equiv$ Valor del índice de eficiencia compuesto de la mejor solución hallada \bar{X} .

$Z^{cur} \equiv$ Valor del índice de eficiencia compuesto de la solución en curso X^{cur} .

$Z^{vec} \equiv$ Valor del índice de eficiencia compuesto de la solución vecina X^{vec} .

$$\Pi(Z^{cur}, Z^{vec}) \equiv \text{Umbral de probabilidad.}$$

Figura 6.12: Pseudocódigo del procedimiento SA-HeufJSP.

Un concepto importante de esta metaheurística es el umbral de probabilidad $\Pi(Z^{cur}, Z^{vec})$, que toma valores entre 0 y 1. Para encontrar el valor del umbral hay que aplicar la ecuación (Ec. 6.17).

$$\Pi = e^{\left(\frac{Z^{vec} - Z^{cur}}{T} \right)} \quad (\text{Ec. 6.17})$$

Se define como la probabilidad de aceptar soluciones peores que la solución actual. Esta aceptación permite escapar de óptimos locales con el objetivo de dirigirse, con mayor o menor demanda de recursos, hacia el óptimo global. Su valor será menor en cuanto mayor sea la diferencia entre la solución vecina y la solución actual o en curso y menor sea el valor de la temperatura T . La temperatura va disminuyendo a medida que el procedimiento SA-HeufJSP va avanzando. La reducción se produce cada IT iteraciones con un parámetro α que multiplica a T . Existe un valor mínimo de la temperatura, que una vez sobrepasado, se devuelve el valor inicial de T , de manera que ayuda a que el valor del umbral de probabilidad sea mayor y aumente la probabilidad de aceptar soluciones peores.

El procedimiento descrito contiene cuatro parámetros que se tendrán que calibrar. Para ello, se empleará el procedimiento CALIBRA, al igual que en la metaheurística GRASP. En el Anexo 2 se puede encontrar más información sobre CALIBRA. Los cuatro parámetros a calibrar son:

- Temperatura T que participa en la función del umbral de probabilidad.
- Parámetro α que reduce la temperatura ($T = \alpha \cdot T$).
- Valor del intervalo de iteraciones (IT) en que se aplicará el parámetro α a la temperatura T .
- Temperatura mínima (T^{\min}) que puede llegar a alcanzar la temperatura T .

Para completar la descripción de la metaheurística *Simulated Annealing*, falta definir el proceso para generar una solución vecina. El método seleccionado para generar el vecindario consiste en intercambiar la posición de dos trabajos, consecutivos o no consecutivos. El método es el mismo que el utilizado en el proceso de optimización local del procedimiento GRASP-HeufJSP (Figura 6.9).

6.2.7. GRASPSA-HeufJSP

El último procedimiento propuesto para resolver el problema *flexible Job-Shop* consiste en un procedimiento híbrido que combina las dos metaheurísticas definidas en los anteriores apartados: GRASP y *Simulated Annealing* (SA).

La estructura es la misma que se ha planteado en GRASP-HeufJSP, pero se sustituye la fase de optimización local por SA-HeufJSP. El objetivo principal de esta hibridación es dirigir el algoritmo hacia el óptimo global y no estancarse en óptimos locales, tal y como pasa con GRASP-HeufJSP. En resumen, el procedimiento consiste en encontrar un conjunto de soluciones generadas con azar orientado o no orientado durante un tiempo de cálculo limitado, seleccionar la mejor solución encontrada y aplicarle el procedimiento SA-HeufJSP para encontrar mejores soluciones. La diferencia principal de la metaheurística *Simulated Annealing* con respecto a un proceso de optimización local es que la primera permite seleccionar soluciones peores y así poder escapar de óptimos locales. La orientación del azar se seleccionará en función de la calidad de las soluciones encontradas con GRASP-HeufJSP, después de analizar los resultados del experimento computacional (expuesto en el apartado 7).

El pseudocódigo que sigue el procedimiento GRASPSA-HeufJSP es el siguiente (Figura 6.13):

0. Iniciar a cero el temporizador t , valor del índice de eficiencia compuesto de la mejor solución $\bar{Z} = \infty$ y de la mejor solución encontrada por azar $\bar{Z}^{azar} = \infty$. Definición de las condiciones de fin: t^{azar} y t^{SA} .
1. Mientras $t < t^{azar}$:
 - 1.1. Obtención de una solución por azar orientado o desorientado (en función de la versión deseada).
 - 1.1.1. Ordenar los trabajos con el índice de prioridad IP^{GRASP} para encontrar una secuencia, la cual será la lista de candidatos. Hacer hasta que la lista de candidatos esté vacía:
 - 1.1.1.1. Cálculo del tamaño de la lista restringida (k), así como los rangos de aceptación.
 - 1.1.1.2. Generar un número al azar ($rnd \in [0,1)$) y escoger el trabajo que contenga el valor.
 - 1.1.1.3. Programar de forma óptima las operaciones del trabajo seleccionado con el método del grafo de múltiples etapas. Eliminar el trabajo ya programado de la lista de candidatos.
 - 1.1.2. Calcular el valor del índice de eficiencia compuesto de la solución (Z^{azar}).

- 1.2. Si $Z^{azar} < \bar{Z}^{azar}$, entonces la mejor solución encontrada por azar pasa a ser X^{azar} ($\bar{X}^{azar} = X^{azar}$ y $\bar{Z}^{azar} = Z^{azar}$).
2. Definir la solución actual (X^{cur}) inicial como la mejor solución encontrada por azar: $X^{cur} = \bar{X}^{azar}$ y $Z^{cur} = \bar{Z}^{azar}$. Además, la mejor solución encontrada hasta el momento es igual a la solución actual inicial: $\bar{X} = X^{cur}$ y $\bar{Z} = Z^{cur}$. Mientras $t < (t^{azar} + t^{SA})$:
- 2.1. Generar una solución vecina, X^{vec} , a partir de la solución actual X^{cur} . Calcular el valor del índice de eficiencia compuesto de la solución vecina Z^{vec} .
- 2.2. Proceso de selección de la solución vecina. Si Z^{vec} no es peor que Z^{cur} ($Z^{vec} \leq Z^{cur}$), entonces la solución vecina pasa a ser la solución actual: $X^{cur} = X^{vec}$ y $Z^{cur} = Z^{vec}$. Si además, la solución vecina es mejor que la mejor solución encontrada hasta el momento: $\bar{X} = X^{vec}$ y $\bar{Z} = Z^{vec}$. En caso contrario, se genera un número al azar $rnd \in [0,1]$ y si $rnd < \prod(Z^{cur}, Z^{vec})$, entonces se acepta la solución que es peor que la actual: $X^{cur} = X^{vec}$ y $Z^{cur} = Z^{vec}$. Comprobar la condición de fin, sino se cumple volver al paso 2.1.

Siendo:

$\bar{X} \equiv$ Mejor solución encontrada hasta el momento.

$\bar{X}^{azar} \equiv$ Mejor solución encontrada por azar.

$X^{azar} \equiv$ Solución encontrada por azar.

$X^{cur} \equiv$ Solución en curso o actual.

$X^{vec} \equiv$ Solución vecina de la solución en curso X^{cur} .

$\bar{Z} \equiv$ Valor del índice de eficiencia compuesto de la mejor solución hallada \bar{X} .

$\bar{Z}^{azar} \equiv$ Valor del índice de eficiencia compuesto de la mejor solución por azar \bar{X}^{azar} .

$Z^{azar} \equiv$ Valor del índice de eficiencia compuesto de la solución por azar X^{azar} .

$Z^{cur} \equiv$ Valor del índice de eficiencia compuesto de la solución en curso X^{cur} .

$Z^{vec} \equiv$ Valor del índice de eficiencia compuesto de la solución vecina X^{vec} .

$\prod(Z^{cur}, Z^{vec}) \equiv$ Umbral de probabilidad.

Figura 6.13: Pseudocódigo del procedimiento GRASPSA-HeufJSP.

7. Experimento computacional

Los procedimientos que se han diseñado y presentado en anteriores apartados, tienen que ser comparados con el algoritmo publicado por Corominas *et al.* (2015). La calidad de cada solución obtenida será evaluada mediante el valor de la función objetivo o índice de eficiencia compuesto. Como se trata de un proceso de minimización, una solución será considerada mejor que otra cuando el valor de su índice de eficiencia (Z) sea menor. La expresión $Z_1 < Z_2$ muestra que la solución 1 tiene mejor calidad que la solución 2.

Para tener unos resultados que indiquen la bondad o no de cada uno de los procedimientos, es necesario realizar un extenso experimento computacional. Se define una muestra de 3888 ejemplares, los cuales son los mismos que se utilizaron en el artículo de referencia. Los algoritmos han sido programados en el lenguaje Java y el experimento computacional ha sido ejecutado en cuatro ordenadores de iguales características (Tabla 7.1).

CARACTERÍSTICAS DE LOS EQUIPOS	
Fabricante	Hewlett-Packard (HP)
Sistema Operativo	Windows 7 Enterprise
Procesador	Intel Core i5 CPU 650 @ 3.20GHz
Memoria RAM	4 GB (3.87 GB utilizables)
Tipo de sistema	64 bits

Tabla 7.1: Características técnicas de los equipos utilizados para el experimento computacional.

7.1. Diseño de los ejemplares

Los ejemplares utilizados son generados por la combinación de una serie de características de los datos del problema a resolver y por la participación del azar, que permite obtener ejemplares distintos de una misma combinación. A continuación e igual que en el artículo de referencia (Corominas *et al.* 2015), se define cada uno de los datos y los valores o rangos que pueden tomar para la generación de los ejemplares:

- Número de máquinas m : 10, 20, 30, 40, 50 o 60.
- Número de trabajos n : 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110 o 120.
- Total de operaciones, h_j , del trabajo j ($j = 1, \dots, n$): El número de operaciones de cada trabajo es un entero generado al azar entre 1 y 10 ($U[1,10]$).

- Conjunto de máquinas, M_{jh} , que pueden procesar la operación O_{jh} ($j=1,\dots,n$; $h=1,\dots,h_j$). Existe una probabilidad pr_j , común a todas las operaciones de un mismo trabajo j , para decidir si la máquina i es capaz de realizar la operación O_{jh} . La inclusión de una máquina al conjunto de candidatas se determina mediante un número aleatorio no mayor de 1, si éste es menor que la probabilidad, la máquina podrá ejecutar la operación. Además, se asegura que toda operación pueda ser realizada por alguna máquina. La probabilidad pr_j también se genera al azar, pero considerando dos escenarios: baja polivalencia de las máquinas ($pr_j = U[0.1, 0.4]$) o alta polivalencia de las máquinas ($pr_j = U[0.5, 0.9]$).
- Tiempo de proceso p_{ijh} para realizar la operación O_{jh} en la máquina i ($j=1,\dots,n$; $h=1,\dots,h_j$; $i \in M_{jh}$). Se utilizan tres intervalos sobre la variabilidad de los tiempos de proceso: baja ($U[1, 30]$), media ($U[1, 90]$) y alta ($U[1, 240]$).
- Energía requerida P_{ijh} para procesar la operación O_{jh} en la máquina i ($j=1,\dots,n$; $h=1,\dots,h_j$; $i \in M_{jh}$). Al tratarse de energía eléctrica, la unidad es el kWh. El valor de la energía se genera al azar: $U[1.1 \cdot p_{ijh}, 1.9 \cdot p_{ijh}]$.
- Fecha de disponibilidad r_j del trabajo j ($j=1,\dots,n$), es decir, el momento en que este trabajo puede iniciarse: $U[0, 359]$.
- Coste de la energía: Se fija en 0,06 €/kWh en las horas valle (hv) y 0,15 €/kWh en las horas punta (hp). Se considera que el día tiene 10 horas punta y el resto son horas valle. Cabe añadir que el período de programación, el cual viene dividido en intervalos iguales, se supone que empieza en la primera hora punta.
- Coeficientes β_j , γ_j y δ_j del índice de eficiencia compuesto. Se definen 3 escenarios que distinguen un ratio entre los costes del *tardiness* y del *earliness*: bajo ($U[0.02, 1, 0.9]$), medio ($U[1/9, 1, 0.5]$) y alto ($U[0.4, 1, 0.2]$).
- Fecha de entrega d_j del trabajo j ($j=1,\dots,n$). El mecanismo para definir este dato es más complejo, con la finalidad de dar mayor realismo a los datos generados. Se define *ejem* como un ejemplar en el que se han establecido los datos: m , n , h_j , M_{jh} , p_{ijh} , P_{ijh} , r_j , hv , hp , β_j , γ_j , δ_j y CI_i . Sea *ejemAuxl* un ejemplar auxiliar que comparte los mismos datos que *ejem*, excepto r_j , hv , hp y β_j . En el ejemplar auxiliar *ejemAuxl*, la fecha de disponibilidad y los costes de la energía se fijan a 0 y el parámetro β_j se define como un valor muy grande; se soluciona el ejemplar *ejemAuxl* con la heurística HeufJSP, la función de ordenación del artículo de

referencia y $\zeta = 0$. El objetivo es que se programen los trabajos lo más temprano posible. Con este cálculo se encuentra el instante en que finaliza (e_j) cada trabajo j . Entonces, la fecha de entrega del trabajo j del ejemplar $ejem$ se define con la expresión (Ec. 7.1):

$$d_j = \max(TIM, e_j + \{U[-0.05 \cdot TIM, 2 \cdot TIM]\}) \quad (\text{Ec. 7.1})$$

Dónde $\{x\}$ devuelve el entero más próximo a x y TIM es el valor esperado del tiempo de proceso total del trabajo, el cual se calcula como el producto del número de operaciones y el tiempo de proceso de una operación.

- Ocupación inicial, CI_i , de la máquina i ($i = 1, \dots, m$). Sea $ejem$ el ejemplar a generar donde las máquinas no tienen ocupación inicial, $CI_i = \emptyset$ ($\forall i$), y los valores m , n , h_j , M_{jh} , P_{ijh} , P_{ijh} , r_j , h_v , h_p , β_j , γ_j y δ_j ya han sido definidos. Se crea un segundo ejemplar auxiliar $ejemAux2$ que comparte los mismos datos que $ejem$, excepto r_j . La fecha de disponibilidad se fija a 0 en $ejemAux2$. En primer lugar, se calculan las fechas de entrega d_j con el proceso descrito en el punto anterior. Luego, se resuelve el ejemplar $ejemAux2$ con la heurística HeufJSP y $\zeta = 0.5$. Finalmente, las ocupaciones iniciales de las máquinas del ejemplar $ejem$ se definen como la ocupación final de las máquinas en la solución encontrada en la resolución del ejemplar auxiliar $ejemAux2$.

Todos estos datos dan lugar a 1296 combinaciones distintas, las cuales surgen de 6 valores que puede tomar el número de máquinas, 12 el número de trabajos, 2 escenarios de la polivalencia de las máquinas, 3 escenarios de la variabilidad del tiempo de proceso y 3 escenarios a cerca del ratio de los costes de *tardiness* y *earliness*. La experiencia computacional se realiza con 3888 ejemplares, es decir, 3 ejemplares para cada combinación. Estos ejemplares son los mismos que los propuestos en el artículo de Corominas *et al.* (2015). A parte de éstos, se generan 1296 ejemplares para calibrar los parámetros de los índices de prioridad (un ejemplar para cada combinación). Finalmente, 96 ejemplares más son creados para calibrar los parámetros utilizados en las metaheurísticas (detallado en el apartado 7.2).

7.2. Calibración de los parámetros

Cada uno de los algoritmos presentados incluye uno o varios parámetros que tienen que ser calibrados. Este proceso de calibración permite ajustar el procedimiento resolutivo al tipo de problema y así obtener mejores resultados.

La Calibración General estática y semidinámica contienen los parámetros en el índice de prioridad, que se utiliza en la ordenación de los trabajos según un orden no decreciente. El proceso consiste en aplicar EAGH más un algoritmo directo de optimización, en este caso, *Nelder & Mead*. Como se ha introducido, se generan 1296 ejemplares de calibración, los cuales se obtienen de las 1296 combinaciones que permiten los datos definidos anteriormente, con lo que se llega a tener un ejemplar para cada combinación. Estos ejemplares son distintos a los que posteriormente serán resueltos.

Nelder & Mead necesita inicializar el baricentro y la longitud del hipertetraedro para poder ser ejecutado. Las coordenadas del punto o baricentro no dejan de ser los valores iniciales que toman los parámetros a calibrar. Se decide fijar los valores iniciales, tanto del baricentro como de la longitud, dando más importancia a los conceptos: fecha de vencimiento normalizada para cada trabajo j (D_j) y estimación normalizada de la energía necesaria para procesar el trabajo j (A_j). Los resultados obtenidos a raíz del proceso de calibración en el procedimiento de Calibración General estática son (Tabla 7.2):

RESULTADOS CALIBRACIÓN CGe	
Concepto	Valor
Tiempo de cálculo [días]	17.79
Valor función EAGH φ [€]	$1.9962 \cdot 10^7$
Iteraciones <i>Nelder & Mead</i>	63
Parámetro α^{CGe}	4.9367
Parámetro β^{CGe}	-0.4235
Parámetro γ^{CGe}	0.0775
Parámetro ε^{CGe}	-0.0429

Tabla 7.2: Resultados obtenidos en la calibración del procedimiento Calibración General estática.

En el caso de la Calibración General semidinámica, se fijan los parámetros de los conceptos comunes con el valor obtenido en la Calibración General estática. Los resultados obtenidos con el proceso de calibración se pueden observar en la Tabla 7.3:

RESULTADOS CALIBRACIÓN CGs	
Concepto	Valor
Tiempo de cálculo [días]	40.14
Valor función EAGH φ [€]	$2.0977 \cdot 10^7$
Iteraciones <i>Nelder & Mead</i>	63
Parámetro α^{CGs}	6.6197
Parámetro β^{CGs}	-1.0650
Parámetro γ^{CGs}	0.2877
Parámetro ε^{CGs}	0.4325
Parámetro σ^{CGs}	-0.1057

Tabla 7.3: Resultados obtenidos en la calibración del procedimiento CGs.

Obtenidos los valores de los parámetros, hay conclusiones relevantes que se pueden extraer. En los dos procesos de calibración, el valor más grande se obtiene en el parámetro α , que pondera la fecha de entrega normalizada, y el más pequeño en el parámetro β , el cual pondera la estimación normalizada de la energía necesaria y toma valor negativo. Con esto, la función de ordenación de trabajos prioriza a los trabajos que tienen una fecha de entrega (d_j) menor y los que necesitarán más energía para ser procesados (parámetro negativo), ya que al principio de la programación habrá más disponibilidad de las máquinas. También en ambos casos, el número de operaciones normalizado tiene un impacto pequeño. Finalmente, destacar que el parámetro σ de la Calibración General semidinámica también es negativo.

Los procedimientos de Calibración Individual estática (CIE) y semidinámica (CIS) tienen un funcionamiento totalmente distinto a los dos anteriores. En este caso no son necesarios unos ejemplares de calibrado, ya que en lugar de encontrar el valor de los parámetros para un grupo de ejemplares representativo, lo que se hace es calibrar los parámetros para cada ejemplar que se resuelve obteniéndose, finalmente, su solución. Este método provocará un aumento del tiempo de cálculo por ejemplar. En la Tabla 7.4 se muestra el promedio de cada parámetro obtenido durante la resolución de los ejemplares.

Los promedios muestran que la tendencia es la misma que en los procesos de Calibración General, ya que se continúa dando más peso a la fecha de entrega y mayor penalización a los trabajos con mayor estimación de la energía necesaria (parámetro negativo). No obstante, recordar que son valores promedio y que los parámetros varían adaptándose a cada ejemplar.

RESULTADOS DEL PROCESO DE CALIBRACIÓN Cle y Cls							
C. Indiv. estática 1		C. Indiv. estática 2		C. Indiv. estática 3		C. Indiv. semi.	
α^{Cle_1}	4.3533	α^{Cle_2}	4.4148	α^{Cle_3}	2.6349	α^{Cls}	6.5733
β^{Cle_1}	-1.0124	β^{Cle_2}	-0.9660	β^{Cle_3}	-0.5627	β^{Cls}	-1.0619
γ^{Cle_1}	0.6631	ε^{Cle_2}	0.5938	γ^{Cle_3}	0.3534	γ^{Cls}	0.4559
				ε^{Cle_3}	0.0814	ε^{Cls}	0.2406
						σ^{Cls}	0.1818

Tabla 7.4: Resultados promedio obtenidos en el procedimiento de calibración de Cle y Cls.

Llegados a este punto, sólo falta por definir los procedimientos para calibrar los parámetros de las metaheurísticas. Para ello se ha empleado el software CALIBRA y un conjunto de 96 ejemplares de calibración. Los ejemplares de calibración están obtenidos por la combinación de estos datos:

- Número de máquinas m : 20, 40 o 60.
- Número de trabajos n : 30, 60, 90 o 120.
- Conjunto de máquinas, M_{jh} , que pueden procesar la operación O_{jh} ($j=1,\dots,n$; $h=1,\dots,h_j$). Dos escenarios: baja polivalencia de las máquinas ($pr_j = U[0.1, 0.4]$) o alta polivalencia de las máquinas ($pr_j = U[0.5, 0.9]$).
- Tiempo de proceso p_{ijh} para realizar la operación O_{jh} en la máquina i ($j=1,\dots,n$; $h=1,\dots,h_j$; $i \in M_{jh}$). Se consideran sólo dos intervalos sobre la variabilidad de los tiempos de proceso: media ($U[1, 90]$) y alta ($U[1, 240]$).
- Coeficientes β_j , γ_j y δ_j del índice de eficiencia compuesto. Se consideran 2 escenarios que distinguen un ratio entre los costes del *tardiness* y del *earliness*: bajo ($U[0.02, 1, 0.9]$) y alto ($U[0.4, 1, 0.2]$).

El procedimiento GRASP-HeufJSP sólo incluye un parámetro a calibrar (k'). Este participa en la primera fase de la metaheurística, con lo que sólo se tendrá en cuenta esta parte para la calibración del parámetro. Para reducir el tiempo de calibrado, se supone que el mismo valor del parámetro k' sirve para las dos versiones del azar: orientado y no orientado. Se ha fijado la condición de fin de la primera fase en 2 minutos por ejemplar y el número máximo de iteraciones del software CALIBRA en 100. Los resultados del proceso de calibrado son mostrados en la Tabla 7.5 (se incluye el intervalo que delimita los valores iniciales del parámetro k'):

PROCESO DE CALIBRACIÓN DE GRASP-HeufJSP			
CONDICIONES INICIALES		RESULTADOS	
Condición de fin	2 min	Tiempo de cálculo [días]	14.1
Número máx. iteraciones CALIBRA	100	Iteraciones CALIBRA	100
Total ejemplares de calibrado	96	Valor parámetro k'	0.1
Intervalo fijado parámetro k'	$[0.1, 1]$		

Tabla 7.5: Condiciones iniciales y resultados obtenidos en la calibración de GRASP-HeufJSP.

El procedimiento SA-HeufJSP tiene 4 parámetros que tienen que ser calibrados. Se utilizan los mismos 96 ejemplares de calibración y las mismas condiciones: 2 minutos por ejemplar y un máximo permitido de 100 iteraciones para el software CALIBRA. Con estas condiciones se obtienen los siguientes valores para cada parámetro (Tabla 7.6):

PROCESO DE CALIBRACIÓN DE SA-HeufJSP			
CONDICIONES INICIALES		RESULTADOS	
Condición de fin	2 min	Tiempo de cálculo [días]	14.06
Número máx. iteraciones CALIBRA	100	Iteraciones CALIBRA	100
Total ejemplares de calibrado	96	Valor parámetro T	162.5
Intervalo fijado parámetro T	$[50, 500]$	Valor parámetro α	0.925
Intervalo fijado parámetro α	$[0.7, 0.999]$	Valor parámetro IT	100
Intervalo fijado parámetro IT	$[50, 250]$	Valor parámetro T^{min}	0.7501
Intervalo fijado parámetro T^{min}	$[0.0001, 1]$		

Tabla 7.6: Condiciones iniciales y resultados obtenidos en la calibración de SA-HeufJSP.

En el procedimiento híbrido GRASPSA-HeufJSP se utilizan los mismos valores de los parámetros que en los métodos de resolución GRASP-HeufJSP y SA-HeufJSP (Tabla 7.5 y Tabla 7.6).

7.3. Análisis de resultados

Una vez descritos los procedimientos diseñados para resolver el problema *flexible Job-Shop*, con un tipo de producción *just-in-time* y considerando la discriminación horaria del coste de la energía eléctrica, y calibrados sus respectivos parámetros, el siguiente paso es determinar la calidad de las soluciones obtenidas en cada caso.

Para ello se realiza un amplio experimento computacional con el conjunto de 3888

ejemplares, definidos de forma detallada en el apartado 7.1, mediante la aplicación de cada uno de los métodos resolutivos. La calidad de cada procedimiento se valorará con una comparación directa con el procedimiento diseñado por Corominas *et al.* (2015). No solo se utilizará la función objetivo o índice de eficiencia compuesto para determinar qué procedimiento es mejor (coste total), sino que se tendrán en cuenta otros resultados de cada uno de ellos. Los ejemplares han sido resueltos para todos los procedimientos, incluso el de referencia, en 4 ordenadores idénticos en características.

Se presentan cinco bloques distintos para describir los resultados de los siete procedimientos planteados. En el primer bloque se incluyen los procedimientos de Calibración General: estática y semidinámica. El segundo junta los dos procedimientos de Calibración Individual: estática y semidinámica. Finalmente, hay un bloque distinto para cada uno de los dos métodos basados en las metaheurísticas GRASP y *Simulated Annealing*: GRASP-HeufJSP y SA-HeufJSP, respectivamente. El último bloque incluye el procedimiento híbrido GRASPSA-HeufJSP. En el *Anexo 3* se pueden encontrar tablas adicionales que analizan en detalle y de forma desglosada los resultados obtenidos en cada uno de los procedimientos presentados. El convenio de signos utilizado considera el signo negativo como mejora respecto al procedimiento de referencia HeufJSP(ζ).

7.3.1. Calibración General

La calibración de los parámetros del índice de prioridad de los trabajos en el procedimiento de Calibración General estática ha conllevado un tiempo de cálculo total de casi 18 días. En cambio, para resolver los 3888 ejemplares sólo ha sido necesario emplear 4.2 horas.

Tal y como se ve en la Tabla 7.7, la resolución en promedio de cada ejemplar es mucho más rápida que con el procedimiento de referencia, destinando sólo unos 3.9 segundos, un 94.2% menos. No obstante, las ventajas en cuestiones de tiempo no se ven reflejadas en los costes. Analizando las dos componentes del coste, se aprecia una paridad en los costes energéticos (diferencia relativa cercana a 0%), pero son los costes de *tardiness* y *earliness* los que penalizan, los cuales son un 46% mayores que en el procedimiento HeufJSP(ζ). Destaca la disminución del *earliness* promedio, lo cual conllevará menos costes, y un aumento muy grande del *tardiness*, que se convierte en el culpable de la gran diferencia que hay en costes de *earliness* y *tardiness*. La conclusión es que el procedimiento planteado peca de finalizar los trabajos más tarde que HeufJSP(ζ) y esto provoca que haya un coste total (índice de eficiencia) promedio un 43.7% peor.

CONCEPTOS	HeufJSP(ζ) Corominas <i>et al.</i> (2015)	Calibración General estática	Calibración General semidinámica
Tiempo promedio por ejemplar [s]	67.4	3.9	8.7
Diferencia relativa tiempo promedio	-	-94.2%	-87.0%
Coste total promedio [€]	15795.5	22700.9	23479.8
Diferencia absoluta coste total por ejemplar [€]	-	6905.4	7684.3
Diferencia relativa coste total	-	43.7%	48.6%
Coste energía promedio [€]	779.5	782.0	782.1
Diferencia relativa coste energía	-	0.3%	0.3%
Coste tardiness y earliness promedio [€]	15016.0	21918.9	22697.7
Diferencia relativa coste tardiness y earliness	-	46.0%	51.2%
Tardiness promedio	3.9	5.4	5.4
Earliness promedio	2.1	1.9	1.9

Tabla 7.7: Resultados promedio del procedimiento de Calibración General estática y Calibración General semidinámica comparados con HeufJSP(ζ).

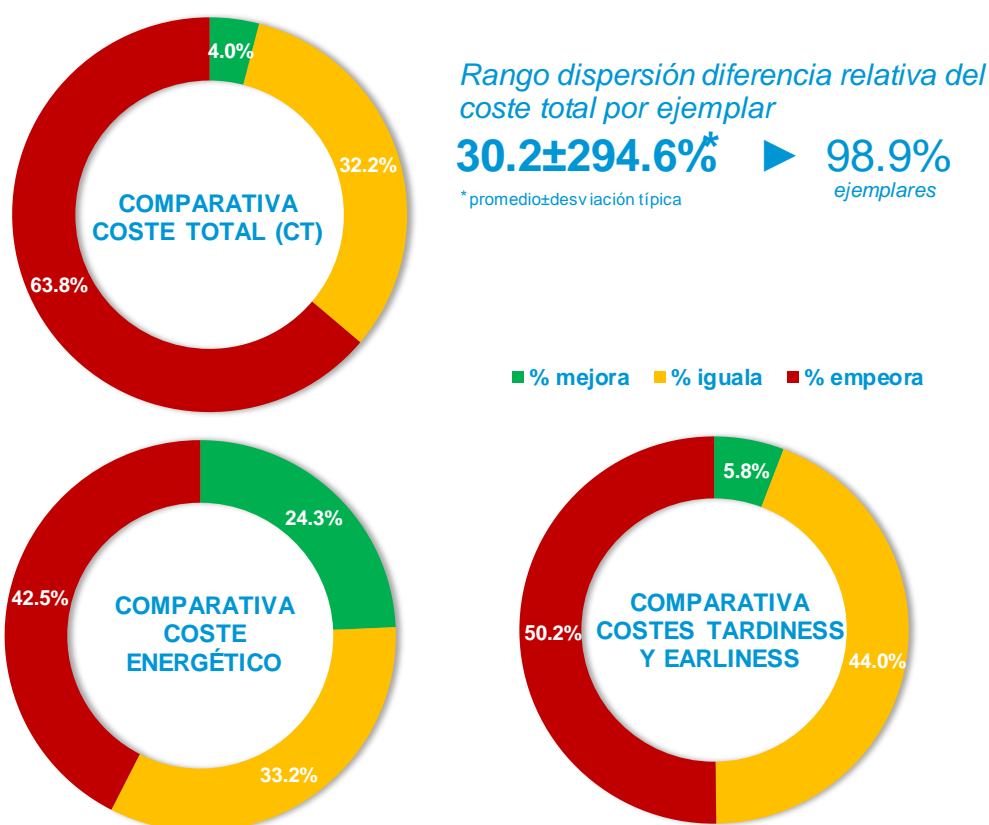


Figura 7.1: Panel de resultados de la Calibración General estática comparando, entre ejemplares, con HeufJSP(ζ). Se muestra el rango de dispersión de la diferencia relativa del coste total por ejemplar.

La Tabla A3.2 del Anexo 3 muestra los valores promedio divididos por grupos de ejemplares en función del ratio n/m (trabajos/máquinas), considerando esta división la más representativa a la hora de clasificar la variedad de los ejemplares. Se ve una tendencia de

los costes a aumentar a medida que el ratio n/m es más grande. Desglosando los resultados y mirando la diferencia relativa del coste total promedio, se ve como el procedimiento tiene un mal comportamiento para todo ratio n/m , llegando a una variación del 75.7% en el ratio con valor (1, 2].

El panel de resultados de la Figura 7.1 confirma todo lo visto en los valores promedios. Sólo en el 4% de los ejemplares testeados se mejora el coste total del procedimiento de referencia. Ni sumando los ejemplares en que se mejora o iguala, se llega al 50%. Se comprueba con claridad que el culpable de estos malos resultados es el coste que hace referencia a la desviación respecto a la fecha de entrega, ya que en sólo el 5.8% de los ejemplares se puede mejorar este concepto. Otro dato que remarca el mal rendimiento de este procedimiento, es el amplio rango de valores que podría tener la diferencia relativa del coste total al ejecutar un único ejemplar. Esta dispersión proporciona mucha incertidumbre en el momento de solucionar un ejemplar, la cual es inadecuada para poder definir un procedimiento como recomendable (ver Figura 7.1).

Como medida de mejora, se diseñó el procedimiento de Calibración General semidinámica que permite recalcular el orden en que se programen los trabajos. Se busca una mejora de la calidad de las soluciones obtenidas, ya que se tendrá en cuenta la variación de la disponibilidad de las máquinas a lo largo del proceso de programación.

En la Tabla 7.7 se expone una comparativa, con respecto al procedimiento de referencia, de los principales conceptos a tener en cuenta, la cual muestra unos valores bastante similares a los obtenidos con Calibración General estática. El tiempo medio por ejemplar es de 8.7 segundos, mayor que el método anterior, pero aún muy lejos de los 67.4 segundos del procedimiento de referencia. Analizando los costes totales, existe un aumento de un 48.6% respecto a HeufJSP(ζ), lo cual da un resultado peor que el procedimiento de Calibración General estática. No obstante, si se compara por ratio n/m (Tabla A3.3) respecto a Calibración General estática (Tabla A3.2), se ve una mejoría en la mayoría de los rangos del coste de *tardiness* y *earliness*. El motivo principal, por el cual el coste total promedio se ve afectado, es el mal resultado de algunos ejemplares en los ratios de valor (4,5], 8 y 12 para el coste de *tardiness* y *earliness*. Por ejemplo, para el ratio de valor 12, el procedimiento de Calibración General estática tiene una diferencia relativa del 45.5% en este coste y, en cambio, el procedimiento semidinámico alcanza el 80.8%.

La Figura 7.2 expone un aumento del porcentaje de ejemplares que mejoran la calidad del procedimiento de Corominas *et al.* (2015). En total son el 5.6% de los ejemplares (+1.6 puntos respecto a la Calibración General estática). Analizando el desglose del coste total, los dos conceptos que lo componen presentan una subida similar. No obstante, los resultados siguen siendo mejorables, ya que en un 61.9% de los ejemplares el

procedimiento de referencia es capaz de obtener costes menores (mayor calidad). Además, en la misma Figura 7.2 se verifica la gran incertidumbre de la calidad que se puede llegar a obtener al solucionar un ejemplar, siendo similar a la vista con el procedimiento de Calibración General estática.

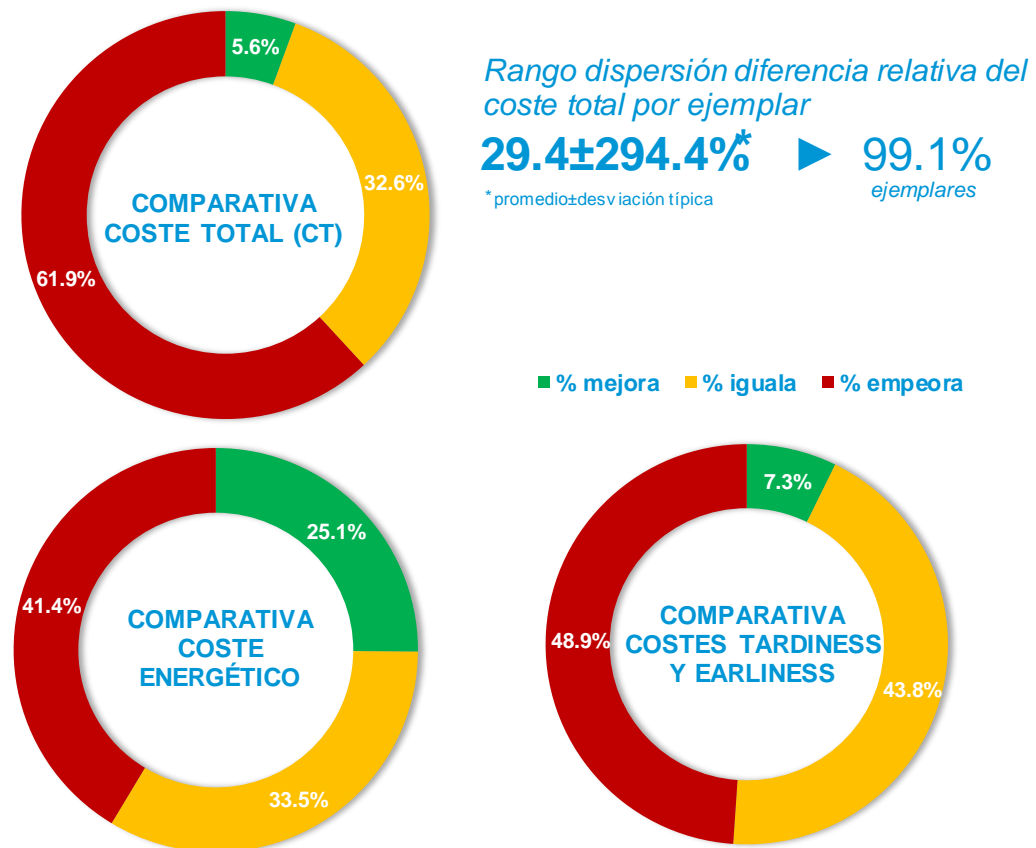


Figura 7.2: Panel de resultados de la Calibración General semidinámica comparando, entre ejemplares, con HeufJSP(ζ). Se muestra el rango de dispersión de la diferencia relativa del coste total por ejemplar.

En conclusión, ambos procedimientos son muy rápidos a la hora de resolver cualquier tipo de ejemplar, pero proporcionan soluciones de baja calidad (alto coste total). Esto tiene efecto en el coste total promedio, el cual es mucho menor con el procedimiento de referencia. Una explicación a estos malos resultados es que los parámetros son calibrados con un conjunto de ejemplares de calibración y los valores obtenidos son aplicados a todos los ejemplares, con lo que no hay una adaptación al tipo de ejemplar. El método HeufJSP(ζ) sí que hace esta adaptación al probar distintos valores de ζ y quedarse con el que proporciona una mejor solución.

7.3.2. Calibración Individual

Los procedimientos basados en la Calibración Individual, estática y semidinámica, buscan mejorar el comportamiento adaptativo del método HeufJSP(ζ) diseñado por Corominas *et al.* (2015) y, a su vez, suplir la carencia que presentan los procedimientos de Calibración General. Se proponen índices de prioridad más complejos y con mayor número de parámetros a calibrarse de forma singular para cada ejemplar.

Empezando por la Calibración Individual estática, tres son los índices de prioridad diseñados (IP^{Cle_1} , IP^{Cle_2} y IP^{Cle_3}) para este procedimiento y que son comparados en la Tabla 7.8. La calibración exclusiva de los parámetros para cada ejemplar provoca un aumento significativo del tiempo de cálculo promedio por ejemplar, quedándose en 517.7, 505.1 y 621.1 segundos, respectivamente. Aunque el aumento del tiempo por ejemplar sea un punto negativo (asumible en la mayoría de entornos industriales reales), se aprecia en la Tabla 7.8 una gran mejora en los costes totales y también en sus dos componentes. Por un lado, para todos los índices de prioridad, la mejora respecto a HeufJSP(ζ) en costes energéticos es muy pequeña, resultado que hace pensar que el procedimiento de referencia ya da buenos resultados en este campo. Por otro lado, la mejora sustancial se produce en una gran disminución de los costes de desviación respecto a la fecha de entrega. Esta reducción de los costes se hace patente en una minoración del *tardiness* promedio.

CONCEPTOS	HeufJSP(ζ) Corominas <i>et al.</i> (2015)	IP^{Cle_1}	IP^{Cle_2}	IP^{Cle_3}	IP^{CIs}
Tiempo promedio por ejemplar [s]	67.4	517.7	505.1	621.1	1552.0
Diferencia relativa tiempo promedio	-	667.8%	649.0%	821.2%	2201.7%
Coste total promedio [€]	15795.5	12894.9	14346.0	12062.0	11666.0
Diferencia absoluta coste total por ejemplar [€]	-	-2900.6	-1449.5	-3589.2	-4129.5
Diferencia relativa coste total	-	-18.4%	-9.2%	-23.6%	-26.1%
Coste energía promedio [€]	779.5	779.1	779.1	776.7	778.0
Diferencia relativa coste energía	-	-0.1%	-0.1%	-0.4%	-0.2%
Coste tardiness y earliness promedio [€]	15016.0	12115.8	13567.0	11285.3	10888.0
Diferencia relativa coste tardiness y earliness	-	-19.3%	-9.7%	-24.8%	-27.5%
Tardiness promedio	3.9	3.4	3.6	3.2	3.2
Earliness promedio	2.1	2.1	2.1	2.1	2.0

Tabla 7.8: Resultados promedio de los tres índices de prioridad pertenecientes a Calibración Individual estática y del índice de Calibración Individual semidinámica, comparándolos con HeufJSP(ζ).

Segregando entre los tres tipos de índices de prioridad que se evalúan con el procedimiento de Calibración Individual estática, se obtiene unos mejores resultados con IP^{Cle_3} . En promedio, el coste total mejora en un 23.6% respecto al procedimiento de referencia, siendo esta reducción menor en el resto de índices (recordar que el convenio de signos que se utiliza considera el signo negativo como una mejora respecto al procedimiento de

referencia). Tal y como se ha dicho en anterioridad, la mayor parte de la reducción del coste total proviene de los costes de *tardiness* y *earliness*. En la Tabla A3.4, Tabla A3.5 y Tabla A3.6 del Anexo 3 se pueden consultar los mismos resultados de la Tabla 7.8, pero desglosados por ratio n/m . Se observa que el índice IP^{Cle_3} mejora al resto de índices en la mayoría de ratios, con excepción de los ratios de valor $(0,1]$ y 9.

El panel de resultados de la Figura 7.3, el cual incluye los porcentajes de mejora, igualdad y empeoramiento de los costes entre pares de ejemplares homólogos, reafirma que el índice IP^{Cle_3} es el que obtiene mejor calidad de las soluciones y mejora de forma sustancial $HeufJSP(\zeta)$. En materia de costes totales, en el 86.39% de los ejemplares se mejora o se iguala la solución, siendo un 47,05% el porcentaje de mejora. Hay que remarcar que el resto de índices también presentan una mejor calidad que el procedimiento de referencia, pero son superados por IP^{Cle_3} en todos los costes. Con los procedimientos de Calibración Individual estática, ha habido una gran disminución de la incertidumbre de la calidad obtenida al ejecutar un único ejemplar, ya que el rango de dispersión de la diferencia relativa del coste total por ejemplar es de menor amplitud. Además, el promedio del rango está en valores de mejora (ver Figura 7.3).

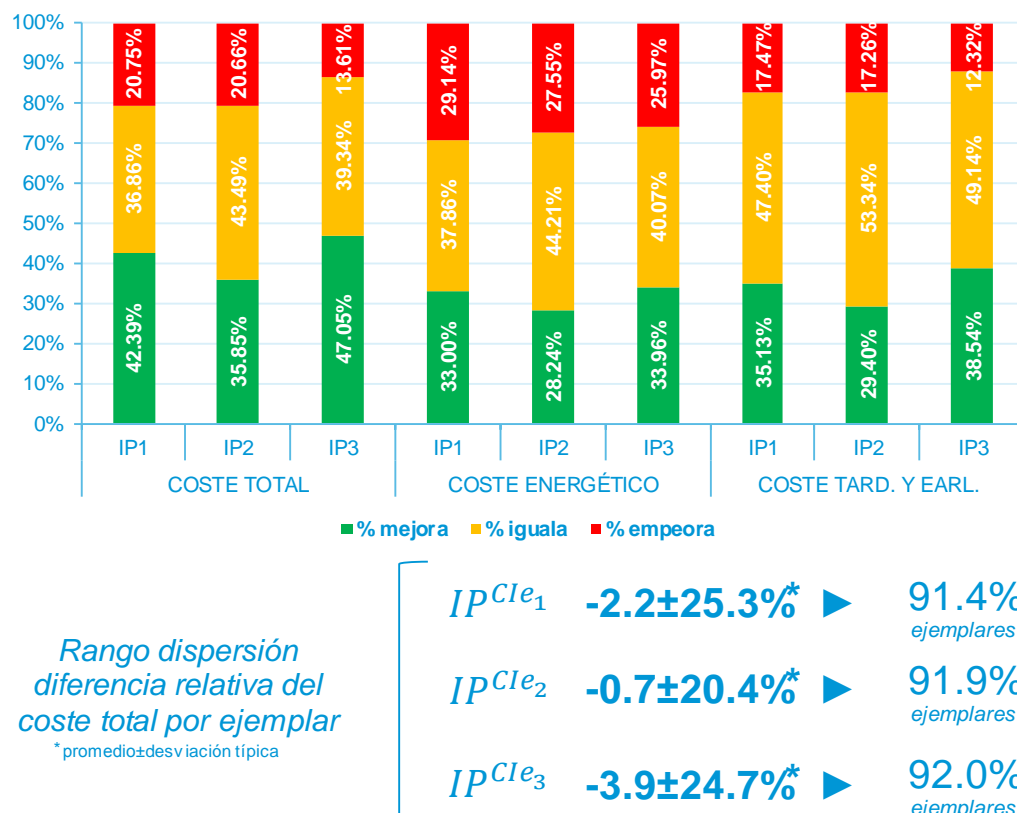


Figura 7.3: Panel de resultados de Calibración Individual estática comparando, entre ejemplares, con $HeufJSP(\zeta)$. Se muestra el rango de dispersión de la diferencia relativa del coste total por ejemplar.

Con el diseño del procedimiento de Calibración Individual semidinámica se marcó el objetivo de intentar mejorar los resultados obtenidos con el índice IP^{Cle_3} de la Calibración Individual estática. La resolución de los 3888 ejemplares ha conllevado un tiempo total de cálculo de 69.8 días, lo que se ha traducido en un tiempo promedio por ejemplar de 1552 segundos. En términos relativos (Tabla 7.8), supone un aumento del 2201.7% respecto al procedimiento de referencia. En promedio, hay una mejora del coste total del 26.1% respecto a HeufJSP(ζ), mejorando también al mejor índice del método de Calibración Individual estática (IP^{Cle_3}). Estas mejoras provienen de una disminución del *earliness*, el cual se traduce en una reducción del 27.5% de su coste vinculado.

Analizando los resultados en más detalle, los cuales se pueden consultar en la Tabla A3.7 del Anexo 3, se puede afirmar que este procedimiento no mejora la solución de referencia en los ejemplares que tienen un ratio n/m inferior o igual a 1. No obstante, el procedimiento de Calibración Individual semidinámica mejora con el aumento de este ratio. Al tratarse de un procedimiento que reordena los trabajos, cuanto mayor sea el número de trabajos, más probabilidad que haya cambios significativos en la secuencia ordenada.

El 88.73% de los ejemplares han superado o igualado la calidad que tenía el procedimiento de Corominas *et al.* (2015), siendo un 45,78% el porcentaje de ejemplares que han mejorado (Figura 7.4). Son más los ejemplares que mejoran o igualan respecto al índice IP^{Cle_3} , pero éste tiene un mayor porcentaje de mejoras.

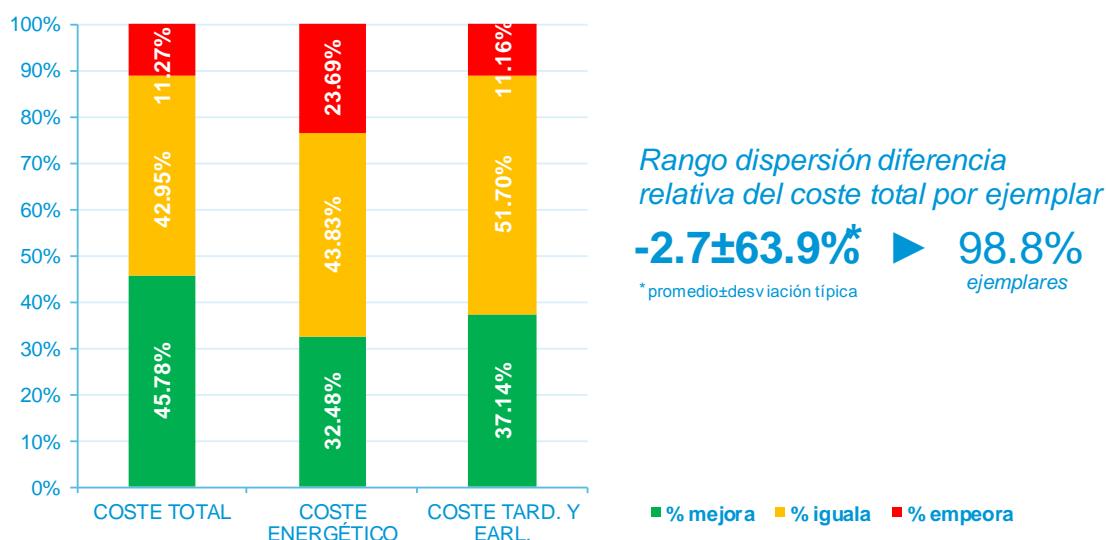


Figura 7.4: Panel de resultados de Calibración Individual semidinámica comparando, entre ejemplares, con HeufJSP(ζ). Se muestra el rango de dispersión de la diferencia relativa del coste total por ejemplar.

En la misma Figura 7.4, se visualiza que con el procedimiento de Calibración Individual semidinámica se vuelve a tener una dispersión de la calidad bastante grande. El hecho de tener esta gran amplitud de la calidad, comparada con el procedimiento de referencia, hace que al ejecutar un ejemplar se pueda obtener una solución muy buena o muy mala. Aunque se consiga un coste total promedio menor, la incertidumbre que se genera al solucionar un ejemplar con Calibración Individual semidinámica hace que no sea conveniente su aplicación.

En conclusión, Calibración Individual estática con el índice de prioridad IP^{Cle_3} es el procedimiento de este apartado con el que se han obtenido mejores resultados y mejor comportamiento. Aun teniendo el procedimiento de Calibración Individual semidinámica un menor número de ejemplares peores y un promedio del coste total menor, el hándicap que supone el mayor tiempo promedio necesario y la incertidumbre que existe al resolver un ejemplar, hacen que sea un procedimiento poco recomendable. Con relación al mayor tiempo de cálculo, destacar también que el tiempo máximo para resolver un ejemplar con Calibración Individual semidinámica ha sido de 3.11 horas, respecto a las 1.07 horas con el índice de prioridad IP^{Cle_3} (ver Gráfico 7.1 y Gráfico 7.2). Otro dato relevante es que con IP^{Cle_3} sólo el 0.1% de los ejemplares supera la hora de cálculo (límite máximo impuesto), en cambio, casi el 13% supera este límite con el procedimiento de Calibración Individual semidinámica (Gráfico 7.1 y Gráfico 7.2: cuadros verdes).

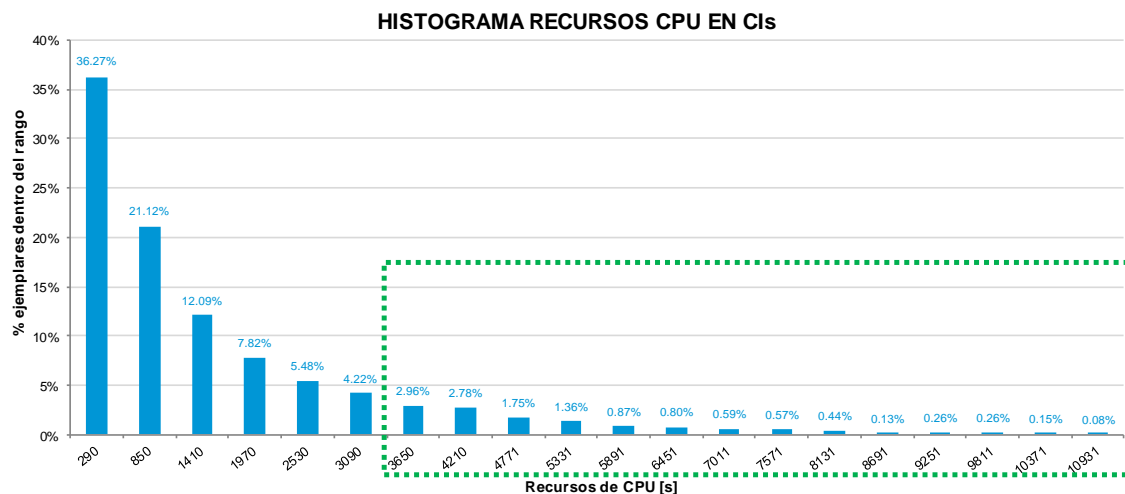


Gráfico 7.1: Histograma de los recursos de CPU necesarios para resolver los ejemplares con el procedimiento de Calibración Individual semidinámica.

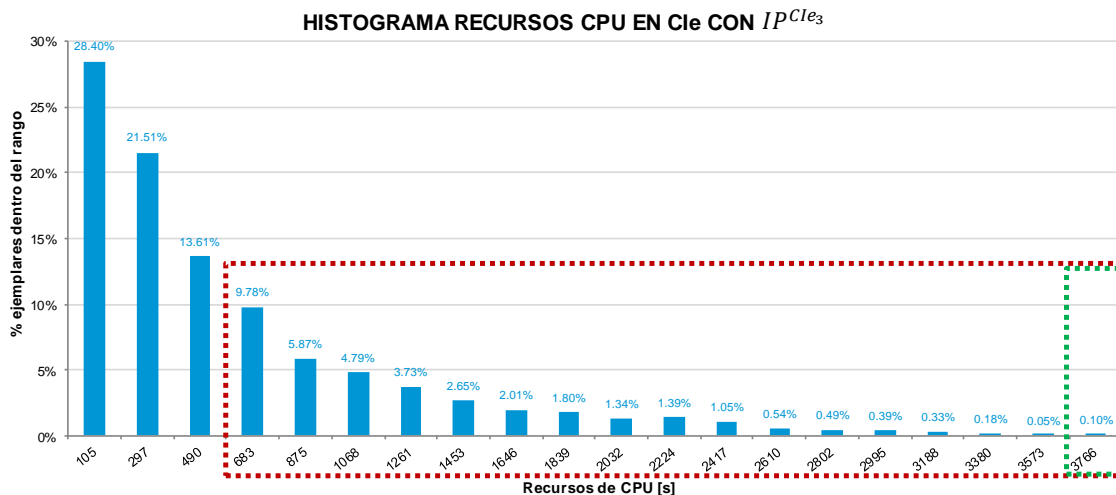


Gráfico 7.2: Histograma de los recursos de CPU necesarios para resolver los ejemplares con el procedimiento de Calibración Individual estática y el índice de prioridad IP^{C1e_3} .

7.3.3. GRASP-HeufJSP

Hasta ahora se han probado dos procedimientos que aplican dos conceptos de optimización de parámetros totalmente distintos, los cuales han conllevado buenos resultados para los procedimientos de Calibración Individual, destacando el índice IP^{C1e_3} , y malos resultados para los procedimientos de Calibración General. La buena calidad en los resultados obtenidos con el procedimiento de Calibración Individual estática con IP^{C1e_3} ha necesitado del consumo de una gran cantidad de recursos de tiempo de CPU, que en definitiva corresponde a tiempo de cálculo. El Gráfico 7.2 representa el histograma de recursos de CPU para el índice de prioridad IP^{C1e_3} .

Con el procedimiento GRASP-HeufJSP y los que se presentarán en los próximos apartados, se marca el objetivo de encontrar un método que permita modificar el tiempo de cálculo y que sea capaz de obtener soluciones de gran calidad en un tiempo no superior a 8 minutos, muy inferior al límite máximo aceptable (1 hora por ejemplar). En el Gráfico 7.2, se remarca con un cuadro rojo la cantidad de ejemplares que no han sido capaces de encontrar una solución en menos de 8 minutos utilizando el procedimiento de Calibración Individual estática y el índice de prioridad IP^{C1e_3} . En definitiva, se quiere reducir el ratio: calidad de la solución obtenida – recursos de CPU necesarios.

Tal y como se ha diseñado el procedimiento GRASP-HeufJSP, este consta de dos fases limitadas por una condición de fin en cada una. La primera de ellas consiste en buscar soluciones con azar orientado o no orientado y, la segunda, consiste en aplicar un proceso de optimización local del tipo ANED con la posibilidad de abortar la optimización si se

cumple una de las dos condiciones definidas. La primera fase está limitada en tiempo a 3 minutos y la segunda está limitada a 5 minutos o también finaliza si no hay ninguna solución vecina que mejora a la solución actual, es decir, que se encuentre un óptimo local. Tal y como está definida esta segunda fase, puede ser que algunos ejemplares se resuelvan en menos de 8 minutos, es decir, que hayan encontrado un óptimo local.

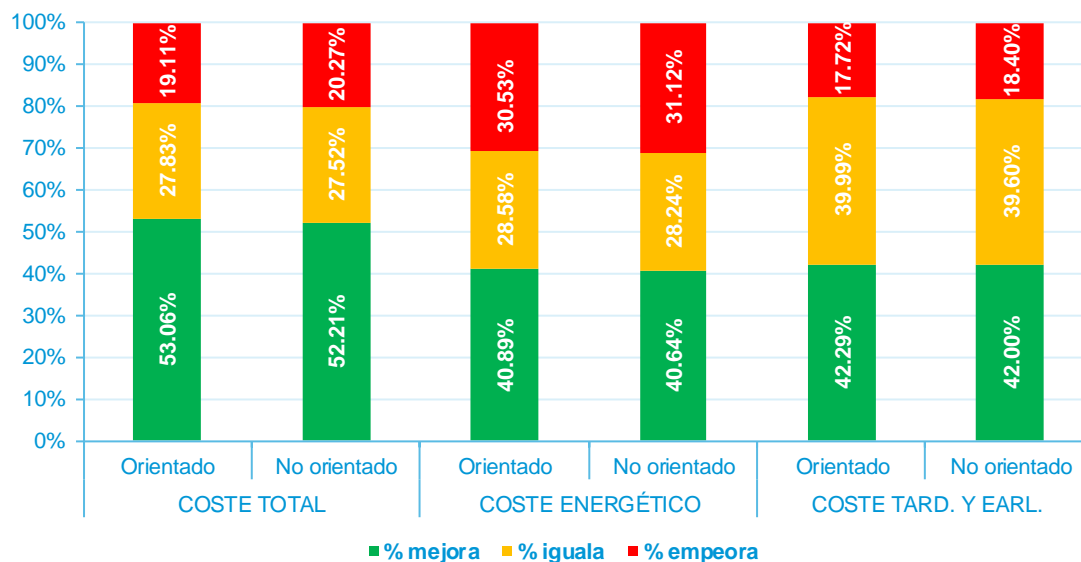
En la Tabla 7.9, se presentan los resultados promedio obtenidos a partir del experimento computacional fruto de la resolución de GRASP-HeufJSP en los dos tipos de orientación del azar: orientado, que es el propuesto por la definición teórica de la metaheurística GRASP, y no orientado, que es una orientación innovadora y presentada en este trabajo.

El tiempo total empleado para resolver los 3888 ejemplares ha sido de 20.5 días para cada uno de los dos casos de orientación del azar, lo cual ha supuesto unos 455.5 segundos de promedio por ejemplar (inferior a los 8 minutos prefijados). Analizando la Tabla A3.8 y la Tabla A3.9 del Anexo 3, se observa que son la mayoría de los ejemplares con un ratio n/m inferior o igual a 1 los únicos que no cumplen con los 8 minutos de límite. Esto significa que terminan la segunda fase del procedimiento porque encuentran un óptimo local, en cambio, el resto se queda en medio proceso de optimización local. Esta limitación puede provocar que no se llegue a un óptimo local, pero es una consecuencia del compromiso entre la calidad de las soluciones obtenidas y los recursos de tiempo de la CPU deseados. Analizando los costes promedios (Tabla 7.9), en ambos casos hay un notorio aumento del coste total promedio con respecto a la heurística HeufJSP(ζ). Por un lado, el coste total promedio del procedimiento GRASP-HeufJSP con azar orientado es un 80.6% mayor que el procedimiento de referencia y, por otro lado, la versión con azar no orientado es un 49.1%. Desglosando los costes, la parte energética mejora ligeramente, pero en cambio, los costes de *tardiness* y *earliness* son ampliamente superiores.

CONCEPTOS	HeufJSP(ζ) Corominas <i>et al.</i> (2015)	GRASP-HeufJSP con azar orientado	GRASP-HeufJSP con azar no orientado
Tiempo promedio por ejemplar [s]	67.4	455.3	455.7
Diferencia relativa tiempo promedio	-	575.3%	575.8%
Coste total promedio [€]	15795.5	28533.7	23555.2
Diferencia absoluta coste total por ejemplar [€]	-	12738.2	7759.7
Diferencia relativa coste total	-	80.6%	49.1%
Coste energía promedio [€]	779.5	776.8	777.0
Diferencia relativa coste energía	-	-0.3%	-0.3%
Coste tardiness y earliness promedio [€]	15016.0	27756.8	22778.2
Diferencia relativa coste tardiness y earliness	-	84.8%	51.7%
Tardiness promedio	3.9	3.5	3.3
Earliness promedio	2.1	2.3	2.3

Tabla 7.9: Resultados promedio de los procedimientos GRASP-HeufJSP con azar orientado y con azar no orientado comparados con HeufJSP(ζ).

El panel de resultados de la Figura 7.5 enseña que con las dos orientaciones del azar se ha mejorado e igualado la solución del procedimiento de referencia en un porcentaje similar, alrededor del 52.5%. Si se tienen en cuenta los ejemplares en que se ha igualado la solución de referencia, se mantiene el empate entre ambas versiones del azar.



Rango dispersión
diferencia relativa
del coste total
por ejemplar
* promedio \pm desviación típica

Orientado	0.8\pm81.0%*	▶	96.4% ejemplares
No orientado	-2.8\pm35.0%*	▶	87.7% ejemplares

Figura 7.5: Panel de resultados de GRASP-HeufJSP comparando, entre ejemplares, con HeufJSP(ζ). Se muestra el rango de dispersión de la diferencia relativa del coste total por ejemplar.

Llegados a este punto, sorprende que las dos versiones del procedimiento GRASP-HeufJSP tengan unos porcentajes de mejora, en número de ejemplares, superiores al procedimiento de Calibración Individual estática con el índice de prioridad IP^{Cle_3} y, a su vez, el coste total promedio sea mucho peor que HeufJSP(ζ). En la Tabla 7.10, se ha realizado una comparativa entre las dos versiones de GRASP-HeufJSP y el procedimiento de Calibración Individual estática con IP^{Cle_3} , con el fin de encontrar el motivo de los resultados obtenidos con GRASP-HeufJSP. Mientras que con el procedimiento de Calibración Individual estática el promedio del coste total es menor para la mayoría de los ratio n/m , las dos orientaciones de GRASP-HeufJSP presentan un empeoramiento del coste total cuando aumenta el ratio. En la Tabla 7.10 se ve una tendencia ascendente del coste total promedio a medida que aumenta el ratio n/m y, por lo tanto, el mal comportamiento de GRASP-HeufJSP para ratios altos afecta en su coste total promedio. La versión con azar no orientado presenta un menor coste total promedio debido a que su empeoramiento es

bastante menor que el visto con GRASP-HeufJSP con azar orientado. El número de ejemplares es desigual para cada ratio n/m , siendo los ratios de mayor volumen los que mejor comportamiento con GRASP-HeufJSP tienen. De aquí que el porcentaje de ejemplares que mejoran al procedimiento de referencia sea tan alto.

Ratio n/m	HeufJSP(¿)		GRASP ORIENTADO				GRASP NO ORIENTADO				Cle con IP^{Cle_3}			
	Ejemplares	Promedio Coste Total	Promedio Coste Total	Diferencia relativa coste total promedio	Ejemplares mejora	Ejemplares mejora o iguala	Promedio Coste Total	Diferencia relativa coste total promedio	Ejemplares mejora	Ejemplares mejora o iguala	Promedio Coste Total	Diferencia relativa coste total promedio	Ejemplares mejora	Ejemplares mejora o iguala
(0,1]	1134	313.9	249.3	-20.6%	30.4%	99.7%	249.3	-20.6%	30.2%	99.6%	337.1	7.4%	15.1%	93.9%
(1,2]	1134	1156.8	698.5	-39.6%	64.1%	87.3%	697.2	-39.7%	62.9%	85.4%	978.4	-15.4%	43.3%	84.9%
(2,3]	648	3755.2	1995.9	-46.8%	71.9%	77.0%	1945.7	-48.2%	66.8%	71.1%	2841.9	-24.3%	60.6%	82.3%
(3,4]	324	9941.3	5733.0	-42.3%	70.1%	70.1%	5034.9	-49.4%	68.2%	68.2%	8021.6	-19.3%	74.4%	79.9%
(4,5]	162	31211.7	19637.5	-37.1%	74.1%	74.1%	22145.2	-29.0%	69.8%	69.8%	21994.6	-29.5%	80.9%	82.7%
(5,6]	162	38034.5	36531.9	-4.0%	56.8%	56.8%	29046.9	-23.6%	59.3%	59.3%	25738.2	-32.3%	83.3%	84.6%
7	54	77692.5	72758.7	-6.4%	63.0%	63.0%	73723.7	-5.1%	72.2%	72.2%	64763.1	-16.6%	74.1%	74.1%
8	54	80679.2	108335.5	34.3%	51.9%	51.9%	95998.4	19.0%	57.4%	57.4%	66140.3	-18.0%	92.6%	92.6%
9	54	97511.7	264314.9	171.1%	20.4%	20.4%	187992.5	92.8%	33.3%	33.3%	99901.2	2.5%	81.5%	81.5%
10	54	158385.0	396610.2	150.4%	13.0%	13.0%	292450.5	84.6%	20.4%	20.4%	113530.3	-28.3%	77.8%	77.8%
11	54	173604.1	342372.3	97.2%	3.7%	3.7%	320776.8	84.8%	11.1%	11.1%	123024.0	-29.1%	85.2%	85.2%
12	54	206072.7	623271.4	202.5%	7.4%	7.4%	498022.4	141.7%	11.1%	11.1%	150615.6	-26.9%	83.3%	83.3%
TOTAL	3888	15795.5	28533.7	80.6%	53.1%	80.9%	23555.2	49.1%	52.2%	79.7%	12062.0	-23.6%	47.0%	86.4%

Tabla 7.10: Evolución con respecto del ratio n/m de los costes totales, porcentajes de mejora y porcentajes de mejora o igualdad para GRASP-HeufJSP y Calibración Individual estática con IP^{Cle_3} .

En conclusión, la versión con azar no orientado del procedimiento GRASP-HeufJSP es la que proporciona mejores resultados de entre las dos propuestas pertenecientes a este apartado. No obstante, la calidad que proporciona este procedimiento no es la deseada, ya que su comportamiento delante de ejemplares con ratios n/m grandes es malo. Además, la dispersión que puede tener la diferencia relativa del coste total al resolver un ejemplar es mayor que con Calibración Individual estática e índice IP^{Cle_3} .

Con el tiempo de cálculo definido, se ha comprobado que en la mayoría de ejemplares el procedimiento no es capaz de encontrar un óptimo local y, en promedio, no es capaz de mejorar al procedimiento de referencia. En otras palabras, los 8 minutos de cálculo son insuficientes para GRASP-HeufJSP. Para comprobar que hubiera sucedido si en lugar de 8 minutos se permitiera cumplir con la limitación máxima de una hora, se decide resolver 14 ejemplares, que pertenecen a ratios n/m superiores o iguales a 9, con una limitación del

tiempo total de una hora por ejemplar. Las condiciones de fin de cada fase se definen manteniendo la proporcionalidad, es decir, 22.5 minutos para la primera fase (3/8) y 37.5 minutos para la segunda (5/8). El experimento se ha realizado con la alternativa de azar no orientado.

El Gráfico 7.3 muestra el promedio del coste total de los 14 ejemplares en diferentes instantes del proceso de resolución. En rojo está marcado el límite de 8 minutos prefijados para resolver cada ejemplar en el experimento computacional. En color lila está el límite de tiempo entre el proceso de obtención de soluciones con azar no orientado y el proceso de optimización lineal. El descenso principal del coste total en promedio se encuentra en los primeros 100 segundos de test. Hasta llegar a los 1350 segundos, que marcan el límite de las dos fases de GRASP-HeufJSP, las soluciones van mejorando. Dentro del proceso de optimización local, se aprecia como las mejoras son menos frecuentes. En ninguno de los 14 ejemplares que se han utilizado, se puede asegurar que se haya encontrado un óptimo local, ya que el procedimiento se ha terminado gracias a la condición de tiempo y no la de comprobación de todas las vecinas.

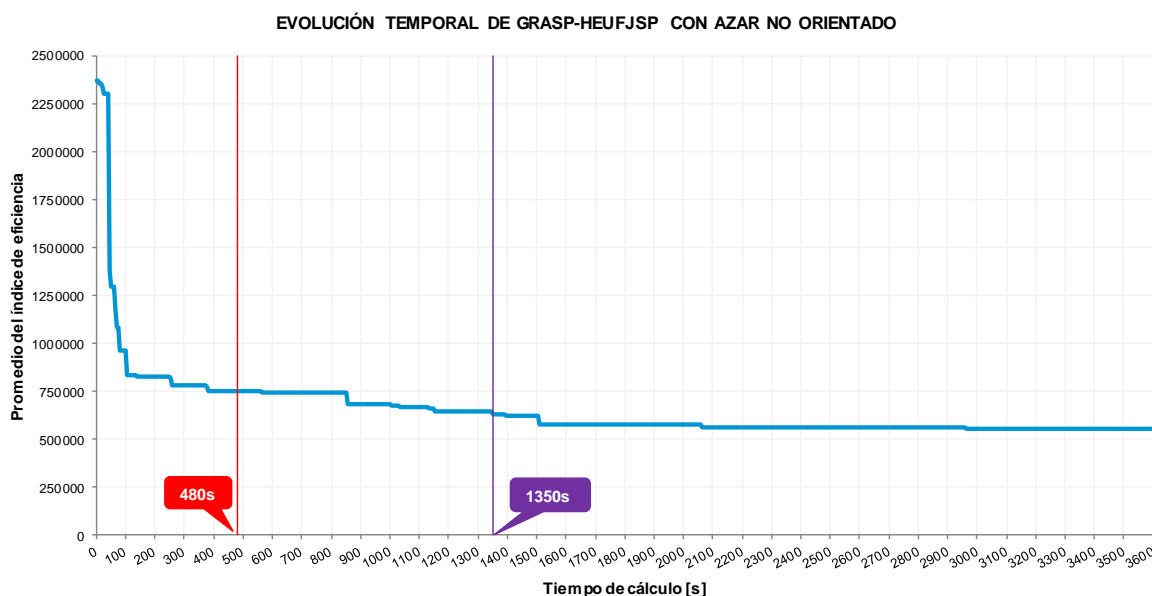


Gráfico 7.3: Experimento, limitado a 1 hora, de GRASP-HeufJSP con azar no orientado teniendo en cuenta el promedio de 14 ejemplares.

En la Tabla 7.11, se presenta la comparativa entre resolver los 14 ejemplares en 8 minutos o en 1 hora. Se puede verificar que la mejora lograda ha sido muy grande en todos los ejemplares, logrando mejorar la solución de referencia en el 78.6% de los ejemplares (en rojo los ejemplares que tienen peor coste total y en verde los que tienen mejor coste). En definitiva, el método GRASP-HeufJSP con azar no orientado es capaz de mejorar las soluciones encontradas con tiempos mayores de 8 minutos, pero la solución se estabiliza bastante a partir de los 2100 segundos (Gráfico 7.3).

COMPARATIVA EXPERIMENTO 1 HORA	HeufJSP(ζ) Corominas <i>et. al.</i> (2015)	GRASP-HeufJSP con azar no orientado (480s)	Dif. Relativa respecto referencia	GRASP-HeufJSP con azar no orientado (3600s)	Dif. Relativa respecto referencia
Ejemplar 1	809764.4	2451136.9	202.7%	1105984.9	36.6%
Ejemplar 2	167133.5	248549.1	48.7%	99691.1	-40.4%
Ejemplar 3	6198.1	23224.5	274.7%	3494.3	-43.6%
Ejemplar 4	65167.7	151087.1	131.8%	47636.6	-26.9%
Ejemplar 5	1407258.9	5390634.9	283.1%	1461448.2	3.9%
Ejemplar 6	112758.9	242485.6	115.0%	92921.1	-17.6%
Ejemplar 7	63262.0	90173.4	42.5%	12710.1	-79.9%
Ejemplar 8	23574.4	36096.4	53.1%	12097.4	-48.7%
Ejemplar 9	58849.5	93150.4	58.3%	38354.4	-34.8%
Ejemplar 10	2063806.4	5530667.1	168.0%	2151847.3	4.3%
Ejemplar 11	2553887.7	6818173.9	167.0%	2448358.7	-4.1%
Ejemplar 12	322014.0	918761.1	185.3%	271648.7	-15.6%
Ejemplar 13	23850.4	63932.2	168.1%	16087.9	-32.5%
Ejemplar 14	48370.9	79522.2	64.4%	15530.8	-67.9%

Tabla 7.11: Comparativa de la mejora experimentada por el coste total antes y después del experimento de evolución temporal en GRASP-HeufJSP con azar no orientado.

7.3.4. SA-HeufJSP

Con el mismo enfoque que GRASP-HeufJSP, se diseña un procedimiento basado en la aplicación de la metaheurística *Simulated Annealing*. En SA-HeufJSP también se define una condición de fin por tiempo, siendo de 8 minutos como en GRASP-HeufJSP.

El tiempo total empleado para ejecutar los 3888 ejemplares ha sido de 21.9 días, lo que ha supuesto un tiempo medio por ejemplar de 485.8 segundos (superior a 480 segundos por la manera como se ha programado el procedimiento, el cual deja terminar una iteración empezada aunque se haya superado el tiempo límite). Observando los resultados de la Tabla 7.12, este procedimiento presenta un coste total promedio que mejora en un 25.3% a HeufJSP(ζ). Esta gran mejora proviene de los costes relacionados con el *tardiness* y *earliness*. Viendo los promedios de estos dos conceptos, hay un empeoramiento del *earliness*, pero la gran ganancia de calidad viene a través de una reducción muy grande del *tardiness* promedio.

En el panel de resultados de la Figura 7.6, se observa que pese a los buenos resultados en promedio, el porcentaje de ejemplares en que se mejora a HeufJSP(ζ) es bastante bajo. SA-HeufJSP, con un tiempo de cálculo de 8 minutos por ejemplar, sólo ha sido capaz de mejorar el 40.6% del conjunto de ejemplares. Si se añade los casos en que se iguala la

solución obtenida con HeufJSP(ζ), se llega sólo a un 70.4%.

CONCEPTOS	HeufJSP(ζ) Corominas <i>et al.</i> (2015)	SA-HeufJSP
Tiempo promedio por ejemplar [s]	67.4	485.8
Diferencia relativa tiempo promedio	-	620.5%
Coste total promedio [€]	15795.5	11792.7
Diferencia absoluta coste total por ejemplar [€]	-	-4002.8
Diferencia relativa coste total	-	-25.3%
Coste energía promedio [€]	779.5	782.5
Diferencia relativa coste energía	-	0.4%
Coste tardiness y earliness promedio [€]	15016.0	11010.1
Diferencia relativa coste tardiness y earliness	-	-26.7%
Tardiness promedio	3.9	2.8
Earliness promedio	2.1	2.4

Tabla 7.12: Resultados promedio de SA-HeufJSP comparado con HeufJSP(ζ).

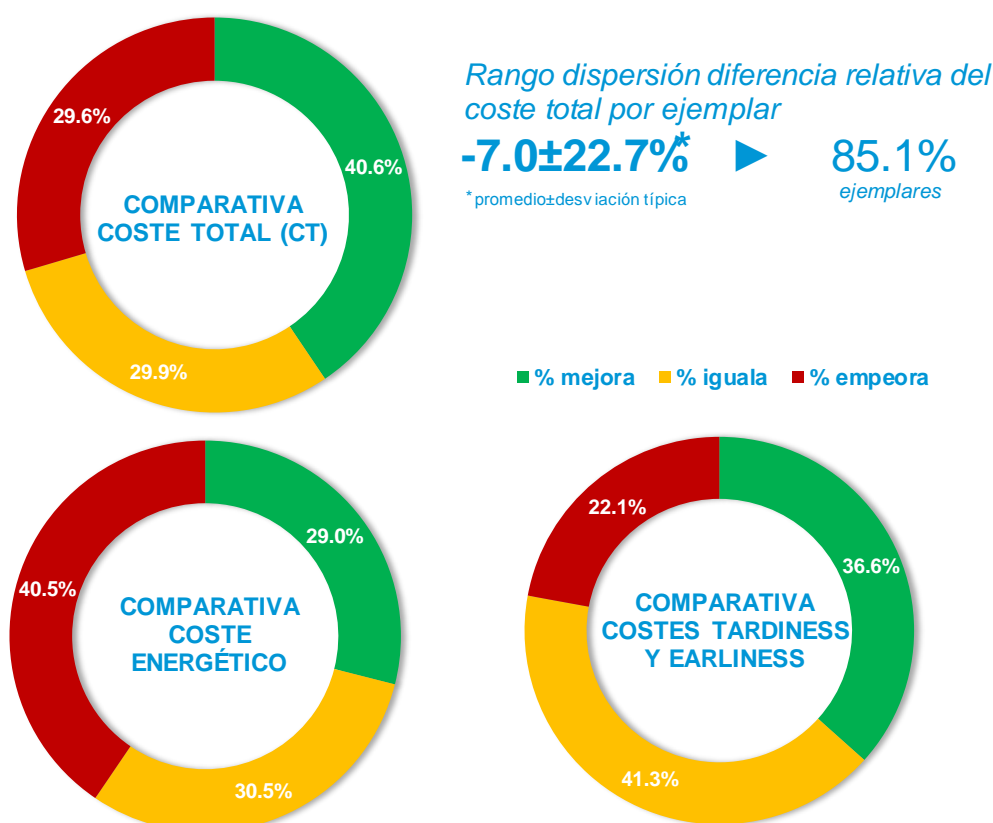


Figura 7.6: Panel de resultados de SA-HeufJSP comparando, entre ejemplares, con HeufJSP(ζ). Se muestra el rango de dispersión de la diferencia relativa del coste total por ejemplar.

Una explicación a este porcentaje de mejora tan bajo y, al mismo tiempo, una calidad un 25.3% mejor que el procedimiento de Corominas *et al.* (2015) se puede encontrar con el

rango de dispersión de la diferencia relativa del coste total al ejecutar un único ejemplar. Se trata del rango de menor amplitud y menor promedio encontrado hasta el momento. Este rango indica que lo más probable es que como mucho se tenga un 15.7% de empeoramiento de la calidad, muy por debajo del 32.2% de GRASP-HeufJSP con azar no orientado o el 20.8% del procedimiento de Calibración Individual estática con IP^{Cle_3} . En consecuencia, estos datos indican que los ejemplares que empeoran la solución del procedimiento de referencia, tienen una calidad no tan mala (menor diferencia relativa).

SA-HeufJSP tiene un comportamiento distinto al procedimiento basado en la metaheurística GRASP. Mientras este último tiene problemas para encontrar buenas soluciones en ratios n/m altos, el método SA-HeufJSP tiene un porcentaje de mejora muy equilibrado para diferentes rangos de este ratio (Tabla A3.10 del Anexo 3). Es decir, SA-HeufJSP es más robusto y/o estable delante de ejemplares de distintas características.

En conclusión, el procedimiento SA-HeufJSP presenta grandes resultados en el coste total promedio, siendo los mejores valores logrados hasta el momento. Asimismo, el porcentaje de mejora de los ejemplares respecto al procedimiento de referencia es muy bajo comparado con otros procedimientos analizados con anterioridad. No obstante, se ha demostrado que los ejemplares que no consiguen mejorar a la solución de referencia tienen una calidad no demasiado distante a ésta.

El bajo porcentaje de ejemplares en los que se mejora HeufJSP(ζ) es un indicio que la condición de fin fijada en 8 minutos se convierte en un limitante para poder encontrar mejores soluciones. Para ver el efecto de la limitación del tiempo en los resultados obtenidos, se realiza un estudio de evolución temporal. Se seleccionan los 14 peores ejemplares para este procedimiento. El Gráfico 7.4 muestra el promedio del coste total de los ejemplares en diferentes instantes del proceso de evolución temporal. La línea roja indica el límite de calidad que se obtendría con la resolución a 8 minutos. Se fija el tiempo máximo por ejemplar en una 1 hora (límite máximo definido). La mejora de la solución es más pronunciada en los primeros segundos, pero se aprecia como a partir de los 480 segundos aún sigue con una pendiente decreciente considerable. Además, la mejora es sostenida y hace pensar que un mayor tiempo de cálculo, conllevaría mejorar aún más las soluciones logradas.

En la Tabla 7.13, se presenta la comparativa entre resolver los 14 ejemplares en 8 minutos o en 1 hora. Se puede verificar que la mejora lograda ha sido grande en la mayoría de ejemplares, logrando mejorar la solución de referencia en el 57.1% de los ejemplares (en rojo los ejemplares que tienen peor coste total y en verde los que tienen mejor coste). El porcentaje de mejora es inferior que con el experimento de evolución temporal realizado con GRASP-HeufJSP. Conviene destacar que en los ejemplares 2 y 5, el coste total obtenido

después del experimento con limitación de 1 hora es peor que con el cálculo limitado a 8 minutos. Este hecho remarca la importancia que tiene el azar en este tipo de procedimientos.

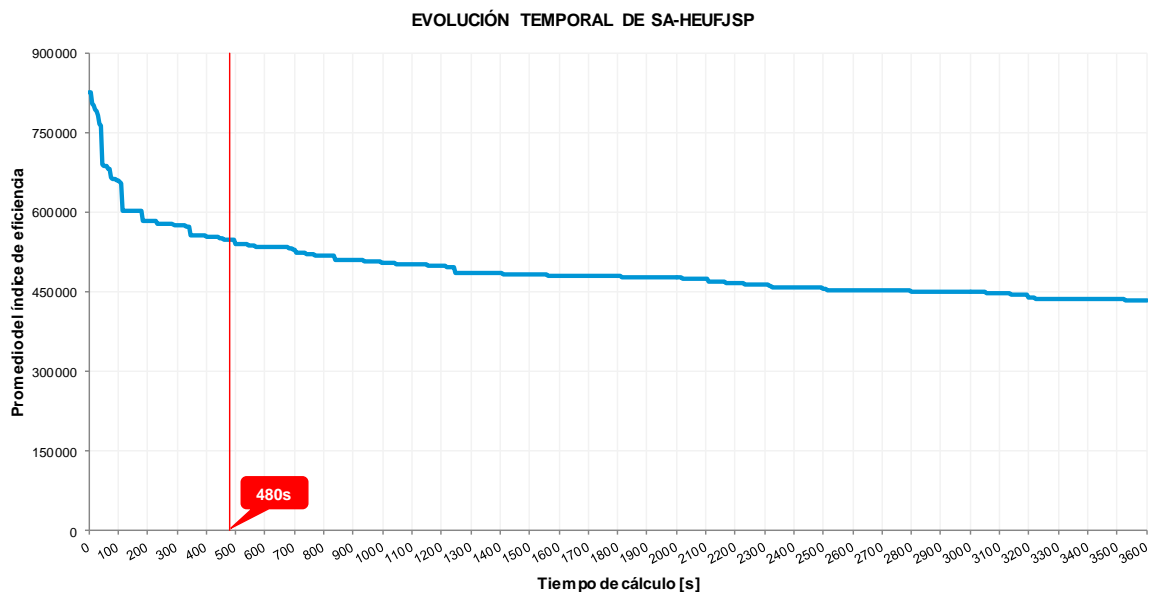


Gráfico 7.4: Evolución temporal de SA-HeufJSP teniendo en cuenta el promedio de 14 ejemplares. La línea roja marca la condición de fin del experimento computacional.

COMPARATIVA EXPERIMENTO 1 HORA	HeufJSP(ζ) Corominas <i>et. al.</i> (2015)	SA-HeufJSP (480s)	Dif. Relativa respecto referencia	SA-HeufJSP (3600s)	Dif. Relativa respecto referencia
Ejemplar 1	982077.4	1392797.7	41.8%	927767.2	-5.5%
Ejemplar 2	751944.8	1125840.1	49.7%	1149299.4	52.8%
Ejemplar 3	576814.5	916049.7	58.8%	661776.2	14.7%
Ejemplar 4	556565.2	704772.7	26.6%	131543.7	-76.4%
Ejemplar 5	1305425.0	1406422.8	7.7%	1884449.7	44.4%
Ejemplar 6	304208.5	397655.6	30.7%	65582.4	-78.4%
Ejemplar 7	413696.3	497173.1	20.2%	339565.7	-17.9%
Ejemplar 8	332712.4	382766.7	15.0%	347360.3	4.4%
Ejemplar 9	94550.5	141523.5	49.7%	77449.3	-18.1%
Ejemplar 10	45296.7	91748.8	102.6%	57003.9	25.9%
Ejemplar 11	270822.4	316000.7	16.7%	303863.3	12.2%
Ejemplar 12	43576.3	86716.2	99.0%	3827.7	-91.2%
Ejemplar 13	34955.3	65249.4	86.7%	31971.0	-8.5%
Ejemplar 14	112758.9	137216.6	21.7%	87991.9	-21.9%

Tabla 7.13: Comparativa de la mejora experimentada por el coste total antes y después del experimento de evolución temporal en SA-HeufJSP.

7.3.5. GRASPSA-HeufJSP

El procedimiento GRASP-HeufJSP con azar no orientado tiene la capacidad de mejorar hasta el 52.2% de los ejemplares, siendo muy efectivo para ratios n/m bajos. En cambio, a medida que el ratio aumenta, su capacidad de encontrar mejores soluciones decrece de forma considerable. Después del análisis de evolución temporal, se puede tasar su limitación de mejorar las soluciones en unos 35 minutos, que equivalen a 2100 segundos. Por otro lado, el segundo procedimiento metaheurístico presentado, el SA-HeufJSP, destaca por su buen desempeño delante de ejemplares con distintos ratios n/m , su mejora progresiva a lo largo del tiempo (ver Gráfico 7.4) y tener un coste total promedio muy bajo. Como se ha descrito, los procedimientos basados en metaheurísticas y propuestos hasta ahora tienen sus ventajas e inconvenientes. Para juntar las ventajas de las dos metaheurísticas y mitigar sus desventajas, se ha propuesto un tercer procedimiento que se ha llamado GRASPSA-HeufJSP. En GRASP-HeufJSP, el mejor coste total promedio se ha conseguido con el azar no orientado, así pues, es la orientación utilizada en GRASPSA-HeufJSP.

La Tabla 7.14 muestra una comparativa directa con el método HeufJSP(ζ) propuesto por Corominas *et al.* (2015). El tiempo de cálculo se ha limitado en 8 minutos por ejemplar, al igual que en los otros dos procedimientos. Para la resolución de los 3888 ejemplares se ha invertido un total de 21.8 días (el tiempo promedio por ejemplar es mayor que 480 segundos por la manera como se ha programado el procedimiento, *ídem* SA-HeufJSP). El promedio del coste total es un 21.8% mejor que el del procedimiento de referencia, bastante similar a SA-HeufJSP. En este procedimiento se consigue mejorar tanto los costes energéticos, como los relacionados con el *tardiness* y el *earliness*. No obstante, los segundos costes continúan siendo la mayor fuente de mejora.

CONCEPTOS	HeufJSP(ζ) Corominas <i>et al.</i> (2015)	GRASPSA-HeufJSP (azar no orientado)
Tiempo promedio por ejemplar [s]	67.4	484.5
Diferencia relativa tiempo promedio	-	618.6%
Coste total promedio [€]	15795.5	12345.2
Diferencia absoluta coste total por ejemplar [€]	-	-3450.3
Diferencia relativa coste total	-	-21.8%
Coste energía promedio [€]	779.5	778.3
Diferencia relativa coste energía	-	-0.2%
Coste tardiness y earliness promedio [€]	15016.0	11567.0
Diferencia relativa coste tardiness y earliness	-	-23.0%
Tardiness promedio	3.9	2.6
Earliness promedio	2.1	2.3

Tabla 7.14: Resultados promedio de GRASPSA-HeufJSP comparado con HeufJSP(ζ).

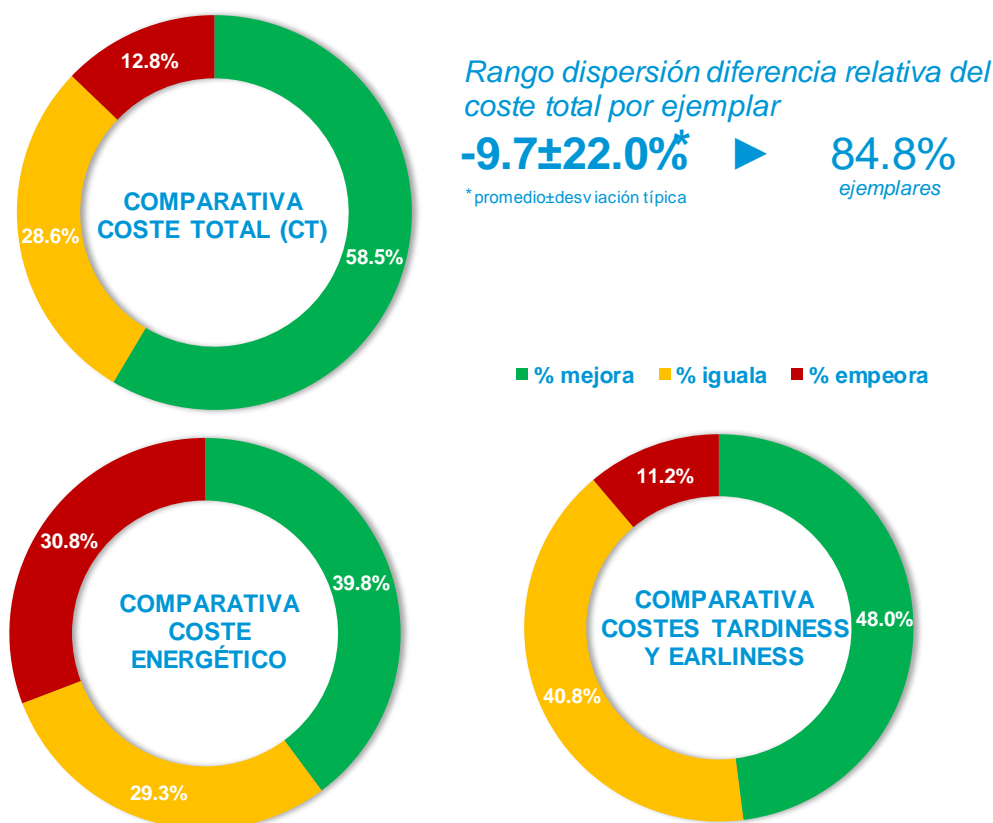


Figura 7.7: Panel de resultados de GRASPSA-HeufJSP comparando, entre ejemplares, con HeufJSP(ζ). Se muestra el rango de dispersión de la diferencia relativa del coste total por ejemplar.

Desglosando por ratio n/m , los porcentajes de mejora de ejemplares y los que se mejora o iguala, se puede observar el gran paso adelante que se ha conseguido con el procedimiento GRASPSA-HeufJSP (Tabla A3.11). Este procedimiento híbrido, que combina las dos metaheurísticas, coge las ventajas de cada una de ellas trabajando por separado. En los ratios bajos, su comportamiento es similar a GRASP-HeufJSP. En cambio, para los ratios altos se asemeja a SA-HeufJSP. Si se tiene en cuenta los valores globales (Figura 7.7), en un 58.5% de los ejemplares resueltos se ha conseguido superar la solución encontrada con HeufJSP(ζ). El valor del porcentaje de ejemplares que han superado o igualado la calidad de Corominas *et al.* (2015) asciende hasta el 87.2%. No obstante, este procedimiento tiene ciertos problemas para alguna ratios n/m , en especial los de valor 9 y 12 (Tabla A3.11). En estos ratios, el coste total promedio no es capaz de mejorar a la solución de referencia y hace pensar que sea necesario un mayor tiempo de cálculo.

Al igual que en los otros procedimientos basados en metaheurísticas, se ha comprobado que la limitación por tiempo puede condicionar el hecho de poder encontrar soluciones de mayor calidad. Para comprobar el efecto de la condición de fin de 8 minutos, impuesta para resolver cada ejemplar, se realiza un estudio de evolución temporal. Se seleccionan 14 ejemplares con mala calidad y se solucionan con una condición de fin de 1 hora. La mayoría

de estos pertenecen a los ratios n/m de valor 9 y 12. Se mantienen las proporciones de GRASP-HeufJSP para las condiciones de fin: la primera fase, que corresponde a GRASP-HeufJSP, está limitada a 22.5 minutos y la segunda a 37.5 minutos, la cual corresponde a SA-HeufJSP. En el Gráfico 7.5 se puede ver una gran mejora durante los primeros segundos y, a posteriori, sigue un descenso aproximadamente lineal y progresivo. Es decir, que dejando el procedimiento durante más tiempo, la solución seguiría mejorando. En el mismo gráfico, se observan las dos características que se han podido ver en los anteriores:

- Fuerte descenso inicial de la evolución temporal de GRASP-HeufJSP (Gráfico 7.3).
- Progresiva mejora de la solución al paso del tiempo, sin que se llegue a estabilizar, de SA-HeufJSP (Gráfico 7.4).

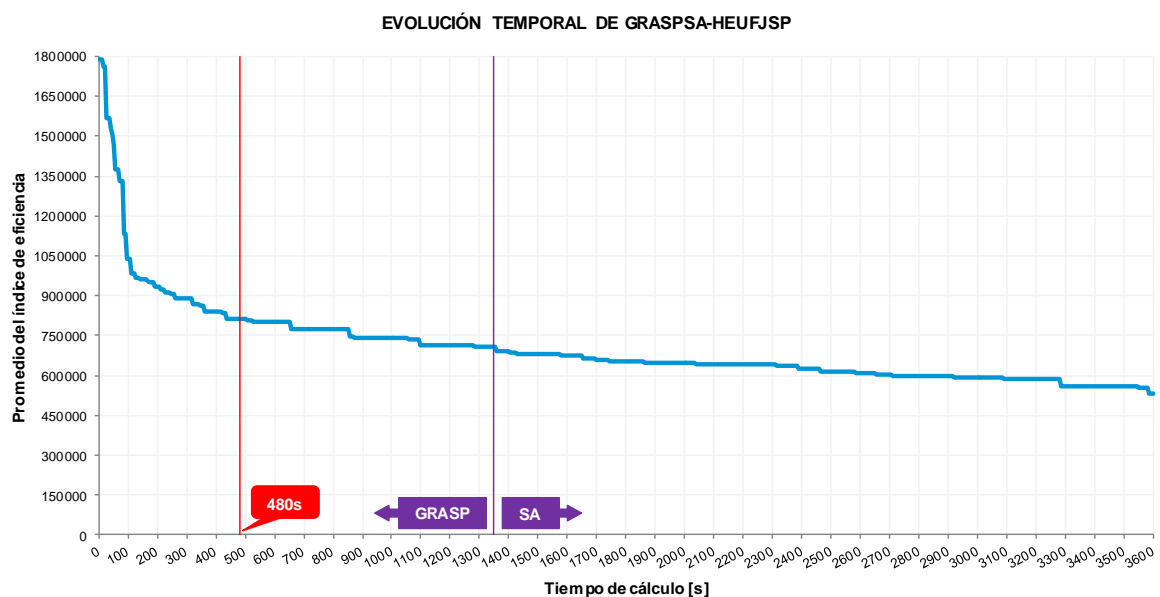


Gráfico 7.5: Evolución temporal de GRASPSA-HeufJSP teniendo en cuenta el promedio de 14 ejemplares. La línea roja marca la condición de fin del experimento computacional y la línea morada delimita las dos fases del procedimiento GRASPSA-HeufJSP.

La Tabla 7.15 muestra la gran mejora del coste total que se ha alcanzado gracias al experimento de evolución temporal. Se ha conseguido mejorar en el 75.6% de los ejemplares probados en el experimento y, en los casos que no se ha mejorado el procedimiento de referencia, se ha logrado una reducción importante de la calidad respecto al cálculo con 8 minutos. Es más, no se ha empeorado ninguna solución con respecto a la resolución con 8 minutos, a diferencia del procedimiento SA-HeufJSP.

COMPARATIVA EXPERIMENTO 1 HORA	HeufJSP(ζ) Corominas <i>et. al.</i> (2015)	GRASPSA- HeufJSP (480s)	Dif. Relativa respecto referencia	GRASPSA- HeufJSP (3600s)	Dif. Relativa respecto referencia
Ejemplar 1	809764.4	1455056.3	79.7%	674160.7	-16.7%
Ejemplar 2	751944.8	1267587.5	68.6%	1045537.7	39.0%
Ejemplar 3	2553887.7	2993261.9	17.2%	2541501.4	-0.5%
Ejemplar 4	982077.4	1293455.6	31.7%	846107.0	-13.8%
Ejemplar 5	247465.5	494001.5	99.6%	139176.9	-43.8%
Ejemplar 6	789306.8	1011879.5	28.2%	642522.8	-18.6%
Ejemplar 7	329063.7	495279.2	50.5%	263465.1	-19.9%
Ejemplar 8	323038.0	451655.4	39.8%	326013.5	0.9%
Ejemplar 9	222895.6	325487.9	46.0%	181704.8	-18.5%
Ejemplar 10	86304.9	187416.8	117.2%	120466.2	39.6%
Ejemplar 11	825940.5	897306.6	8.6%	454026.3	-45.0%
Ejemplar 12	124649.1	173679.2	39.3%	106116.6	-14.9%
Ejemplar 13	93901.1	130888.1	39.4%	68105.9	-27.5%
Ejemplar 14	29491.8	61663.2	109.1%	24221.8	-17.9%

Tabla 7.15: Comparativa de la mejora experimentada por el coste total antes y después del experimento.

8. Resultados finales

8.1. Comparativa entre procedimientos

Una vez descritos los procedimientos y analizados los resultados obtenidos después del experimento computacional realizado, en este apartado se presenta una comparativa entre todos los procedimientos, valorando la calidad de cada uno y escogiendo cuál es el procedimiento que proporciona mejores resultados finales.

Primeramente, se define cual será el usuario habitual de estos algoritmos. En principio, se tratará del jefe de producción o el responsable de organización industrial, el cual deberá planificar la producción y, por consecuencia de esto, deberá programar las operaciones de cada uno de los trabajos a realizar. El problema *flexible Job-Shop* no es aplicable a todo tipo de industrias, sino que serán empresas con una distribución orientada al proceso. Estas empresas se identifican por tener una cierta variedad de productos a fabricar y una demanda baja o intermitente de cada uno de ellos.

En la Tabla 8.1 se visualiza una comparativa entre todos los métodos propuestos y estudiados en el apartado 7.3. Los tiempos y costes son valores promedio del conjunto de 3888 ejemplares utilizados en el experimento computacional. En cuestiones de tiempo, el promedio está dentro del límite de una hora para todos los procedimientos. Sin embargo, se ha demostrado que el algoritmo de Calibración Individual semidinámica ha superado este límite en casi un 13% de los ejemplares (Gráfico 7.1). Por el no cumplimiento del requerimiento de tiempo máximo de cálculo por ejemplar, se considera el procedimiento de Calibración Individual semidinámica como no apto. Valorando la calidad que proporciona cada uno de los procedimientos restantes, se puede ver en la Tabla 8.1 que los métodos de Calibración General empeoran en más de un 40% el promedio del coste total, comparándolo con el procedimiento de referencia (Corominas *et al.* 2015). Igualmente, el mismo problema se reproduce en los procedimientos: GRASP-HeufJSP con azar orientado y GRASP-HeufJSP con azar no orientado. La calidad que se obtiene es peor que la que se obtendría con HeufJSP(ζ). Más aún, estos mismos procedimientos presentan un mal desempeño para ratios n/m altos y tienen un amplio rango de dispersión de la diferencia relativa del coste total al ejecutar un ejemplar, lo que daría mucha incertidumbre al resultado que obtendría el usuario. Recapitulando, estos cuatro procedimientos también se consideran no aptos para su aplicación, ya que no proporcionan una mayor calidad que la obtenida con el procedimiento de referencia. En el caso que se recomendara uno de estos procedimientos, al cabo de un número alto de programaciones variadas, el usuario estaría perdiendo una gran cantidad de dinero comparándolo con HeufJSP(ζ). Esta diferencia no vendría de la

parte energética, dónde las diferencias son mínimas en todos los procedimientos, sino que sería de los costes de retraso (*tardiness*) y adelanto (*earliness*) respecto a la fecha de entrega de los trabajos. En otras palabras, el usuario obtendría una programación que generaría mucho stock de producto acabado y/o un alto número de trabajos que terminaría a posteriori de la fecha de entrega, lo que conllevaría un aumento de las penalizaciones y un empeoramiento de la imagen de la empresa.

COMPARATIVA DE RESULTADOS ENTRE LOS MÉTODOS RESOLUTIVOS PROPUESTOS PARA EL PROBLEMA FLEXIBLE JOB-SHOP	HeufJSP(ζ) Corominas et al. (2015)	Calibración General estática	Calibración General semidinámica	Calibración Individual estática IP^{CLe_3}	Calibración Individual semidinámica	GRASP-HeufJSP orientado	GRASP-HeufJSP no orientado	SA-HeufJSP	GRASPSA- HeufJSP (azar no orientado)
Tiempo total por ejemplar [s]	67.4	3.9	8.7	621.1	1552.0	455.3	455.7	485.8	484.5
Diferencia relativa tiempo total	-	-94.2%	-87.0%	821.2%	2201.7%	575.3%	575.8%	620.5%	618.6%
Coste Total promedio (índice eficiencia) [€]	15795.5	22700.9	23479.8	12062.0	11666.0	28533.7	23555.2	11792.7	12345.2
Diferencia relativa coste total	-	43.7%	48.6%	-23.6%	-26.1%	80.6%	49.1%	-25.3%	-21.8%
% ejemplares mejoran	-	4.0%	5.6%	47.0%	45.8%	53.1%	52.2%	40.6%	58.5%
% ejemplares mejoran o igualan	-	36.2%	38.1%	86.4%	88.7%	80.9%	79.7%	70.4%	87.2%
Costes de energía promedio [€]	779.5	782.0	782.1	776.7	778.0	776.8	777.0	782.5	778.3
Diferencia relativa costes de energía	-	0.3%	0.3%	-0.4%	-0.2%	-0.3%	-0.3%	0.4%	-0.2%
% ejemplares mejoran	-	24.3%	25.1%	34.0%	32.5%	40.9%	40.6%	29.0%	39.8%
% ejemplares mejoran o igualan	-	57.5%	58.6%	74.0%	76.3%	69.5%	68.9%	59.5%	69.2%
Costes tardiness y earliness promedio [€]	15016.0	21918.9	22697.7	11285.3	10888.0	27756.8	22778.2	11010.1	11567.0
Diferencia relativa costes tardiness y earliness	-	46.0%	51.2%	-24.8%	-27.5%	84.8%	51.7%	-26.7%	-23.0%
% ejemplares mejoran	-	5.8%	7.3%	38.5%	37.1%	42.3%	42.0%	36.6%	48.0%
% ejemplares mejoran o igualan	-	49.8%	51.1%	87.7%	88.8%	82.3%	81.6%	77.9%	88.8%
Rango dispersión diferencia relativa coste total por ejemplar	-	30.2± 294.6%	29.4± 294.4%	-3.9± 24.7%	-2.7± 63.9%	0.8± 81.0%	-2.8± 35.0%	-7.0± 22.7%	-9.7± 22.0%

Tabla 8.1: Comparativa de los resultados obtenidos entre los procedimientos diseñados y HeufJSP(ζ).

En resumen, los procedimientos de Calibración Individual estática (IP^{CLe_3}), SA-HeufJSP y GRASPSA-HeufJSP (azar no orientado) son los que presentan mejores resultados. Aunque el procedimiento de Calibración Individual estática haya superado el límite máximo de cálculo en muy pocos ejemplares y de forma muy sutil (tiempo máximo de 1.07 horas), se considera que los tres procedimientos cumplen con el requerimiento del tiempo. Además, la calidad obtenida (coste total) es bastante similar en los tres casos. Por lo tanto, se decide analizar otros conceptos para determinar qué procedimiento es el más recomendado y mejor comportamiento tiene.

Para escoger el mejor procedimiento se elabora una tabla de decisión, en la cual se valorarán cinco conceptos distintos. Al cumplir con el requerimiento de tiempo y tener una calidad similar, siendo estos los conceptos de mayor importancia, se considera que los cinco conceptos tendrán el mismo peso dentro de la cualificación final obtenida con la tabla de decisión. La valoración será del 0 al 5, siendo 0 la peor valoración y 5 la mejor. Los conceptos que se tienen en cuenta se enumeran a continuación:

- ❶ Capacidad del usuario de definir el tiempo de cálculo deseado.
- ❷ Capacidad del procedimiento de mejorar la calidad a mayor tiempo de cálculo.
- ❸ Rango de dispersión de la diferencia relativa del coste total al ejecutar un ejemplar.
- ❹ Porcentaje de ejemplares que han mejorado la calidad de HeufJSP(ζ).
- ❺ Porcentaje de ejemplares que han mejorado o igualado la calidad de HeufJSP(ζ).

Según los resultados de la tabla de decisión (Tabla 8.2), se obtiene que GRASPSA-HeufJSP con azar no orientado es el procedimiento más recomendable y con mejor comportamiento global. De los tres procedimientos, se trata del que tiene una mejora de la calidad inferior (21.8%), pero son los otros conceptos los que lo hacen destacar. Es un procedimiento en el que el usuario será capaz de modificar el tiempo de cálculo, a diferencia de Calibración Individual estática. Con los estudios de evolución temporal, se ha demostrado que GRASPSA-HeufJSP mejora más y en mayor porcentaje de ejemplares que comparado con SA-HeufJSP (Gráfico 7.4 y Gráfico 7.5). El rango de dispersión de la diferencia relativa del coste total por ejemplar es el mejor encontrado, además de tener menor amplitud, tiene el promedio más bajo. Finalmente, también sobresale en el porcentaje de ejemplares que han mejorado la calidad en el experimento computacional.

TABLA DE DECISIÓN	❶	❷	❸	❹	❺	TOTAL
Calibración Individual estática (IP^{CIE_3})	0	0	3	3	4	2.0
SA-HeufJSP	5	4	4	2	3	3.6
GRASPSA-HeufJSP (azar no orientado)	5	5	5	4	4	4.6

Tabla 8.2: Tabla de decisión para seleccionar el mejor procedimiento diseñado.

El objetivo principal, fijado inicialmente, consiste en encontrar la programación de una serie de trabajos optimizando el consumo energético del proceso, reduciendo el stock de producto acabado y evitando el retraso de las entregas, es decir, fomentando el *just-in-time*. Las programaciones de operaciones suelen tener un periodo de aplicación de entre 1 a 6 semanas, pudiendo ser más o menos en función de la voluntad o necesidad del usuario. Es decir, una hora de cálculo para obtener la programación de como mínimo una semana, no es un tiempo desproporcionado y perfectamente aceptable por el usuario. Por esto, el tiempo de cálculo en la utilización de la metaheurística recomendada, GRASPSA-HeufJSP, puede ser un parámetro que decida el usuario, avisándole previamente que a mayor tiempo se pueden obtener mejores resultados. Eso sí, se tendría que limitar o recomendar al mismo usuario que la condición de fin sea de un mínimo de 8 minutos, la misma condición que se ha experimentado en este trabajo y se ha comprobado que permite obtener soluciones de alta calidad.

Encontrado el procedimiento que en general tiene un mejor comportamiento, se decide analizar y determinar qué procedimiento es el mejor para cada ratio n/m . Este método de estudio permite adaptar el procedimiento a utilizar en función del número de trabajos y máquinas del ejemplar, con lo que se obtendrá mayor calidad de las soluciones. Para este análisis no se tendrán en cuenta los procedimientos Calibración General estática, Calibración General semidinámica y Calibración Individual semidinámica. En la Tabla 8.3 se observa como en la mayoría de ratios domina el procedimiento GRASPSA-HeufJSP (azar no orientado). En los ratios n/m de valor 8 a 10, es el procedimiento SA-HeufJSP el que proporciona mayor calidad. El algoritmo de Calibración Individual Estática con índice IP^{Cle_3} es el que da menor coste total promedio para el mayor valor del ratio n/m , muy lejos del que dan otros procedimientos. Por último, los algoritmos basados en la metaheurística GRASP no son capaces de destacar en ningún rango del ratio.

Ratio n/m	Coste total promedio [€]					Mejor procedimiento
	Calibración Individual estática (IP^{Cle_3})	GRASP-HeufJSP orientado	GRASP-HeufJSP no orientado	SA-HeufJSP	GRASPSA-HeufJSP (azar no orientado)	
(0,1]	337.13	249.31	249.32	249.96	249.30	GRASPSA-HeufJSP
(1,2]	978.38	698.53	697.18	693.60	672.66	GRASPSA-HeufJSP
(2,3]	2841.87	1995.93	1945.66	1852.32	1756.98	GRASPSA-HeufJSP
(3,4]	8021.64	5733.01	5034.91	4787.09	4250.78	GRASPSA-HeufJSP
(4,5]	21994.64	19637.52	22145.20	20963.95	15326.82	GRASPSA-HeufJSP
(5,6]	25738.22	36531.95	29046.93	23388.81	18943.88	GRASPSA-HeufJSP
7	64763.08	72758.70	73723.70	62349.42	49638.71	GRASPSA-HeufJSP
8	66140.32	108335.52	95998.42	53376.32	58813.23	SA-HeufJSP
9	99901.20	264314.89	187992.46	94432.53	100824.70	SA-HeufJSP
10	113530.25	396610.21	292450.53	105834.00	132004.99	SA-HeufJSP
11	123024.04	342372.34	320776.77	129324.26	122167.04	GRASPSA-HeufJSP
12	150615.62	623271.40	498022.37	199931.47	256647.22	Calibración Individual estática

Tabla 8.3: Selección del procedimiento de mayor calidad para cada ratio n/m .

Con el experimento computacional se ha llegado a la conclusión que el mejor método testado es GRASPSA-HeufJSP, con una calidad muy superior al procedimiento de Corominas *et al.* (2015). Estos buenos resultados se han conseguido con sólo 8 minutos de cálculo por ejemplar. Ya se ha comprobado en el estudio de evolución temporal que a mayor tiempo de cálculo, este procedimiento es capaz de mejorar las soluciones encontradas, de manera que la diferencia con el procedimiento de referencia aún sería mayor. Esta sería una forma eficaz de encontrar soluciones de mayor calidad, pero aumentando de forma considerable el tiempo de cálculo. Otra manera para obtener mejores resultados es aplicar la técnica de cóctel de heurísticas, la cual fue propuesta por Corominas *et al.* (2012).

Consiste en aplicar varios procedimientos seguidos y quedarse con la solución del procedimiento que mejor resultado logre. En este caso se hará un cóctel de heurísticas para comprobar la variación de los resultados si se combina un procedimiento existente con otro. Los procedimientos de Calibración General no se tienen en cuenta en este análisis por su mala calidad. Tampoco se evalúan los procedimientos de Calibración Individual, ya que su tiempo de cálculo es una incógnita a priori y combinando estos procedimientos con otros se podría superar el límite de una hora. Por lo tanto, se combinarán dos a dos los procedimientos basados en metaheurísticas. Se utilizan los mismos resultados obtenidos con el experimento computacional, de ahí que cada combinación requiera 16 minutos por ejemplar, teniendo aún margen para alcanzar el tiempo límite.

Los resultados obtenidos del análisis de cóctel de heurísticas por combinación de dos procedimientos se resumen en la Tabla 8.4. Como era de esperar, la combinación de los dos mejores procedimientos en el análisis del experimento computacional, SA-HeufJSP y GRASPSA-HeufJSP, ha sido la que ha proporcionado mejores resultados. El coste total promedio ha disminuido en un 37.5% respecto al procedimiento de referencia. Más aún, el número de ejemplares en que se ha mejorado o igualado la solución de HeufJSP(ζ) asciende a un 89.9%, siendo un 60.6% el porcentaje de ejemplares que han mejorado. Añadir también, que el rango de dispersión de la diferencia relativa del coste total al ejecutar un único ejemplar es el mejor encontrado, teniendo el menor promedio. Lo más importante de todo, es que el cóctel de heurísticas de SA-HeufJSP + GRASPSA-HeufJSP ha permitido aumentar la calidad de las soluciones logradas consumiendo sólo 16 minutos, 8 minutos más que con el procedimiento GRASPSA-HeufJSP.

CÓCTEL DE HEURÍSTICAS COMBINANDO PROCEDIMIENTOS METAHEURÍSTICOS DOS A DOS	HeufJSP(ζ) Corominas et al. (2015)	GRASP-HeufJSP (orientado) + GRASP- HeufJSP (no orientado)	GRASP-HeufJSP (orientado) + SA- HeufJSP	GRASP-HeufJSP (orientado) + GRASPSA-HeufJSP	GRASP-HeufJSP (no orientado) + SA- HeufJSP	GRASP-HeufJSP (no orientado) + GRASPSA-HeufJSP	SA-HeufJSP + GRASPSA-HeufJSP
Coste total promedio [€]	15795.5	21969.6	11195.5	12064.4	11185.8	12084.2	9868.8
Diferencia absoluta coste total por ejemplar [€]	-	6174.1	-4600.0	-3731.1	-4609.7	-3711.3	-5926.7
Diferencia relativa coste total	-	39.1%	-29.1%	-23.6%	-29.2%	-23.5%	-37.5%
Porcentaje de ejemplares que mejoran	-	57.4%	59.6%	63.1%	58.4%	62.6%	60.6%
Porcentaje de ejemplares que mejoran o igualan	-	86.0%	88.6%	91.8%	87.4%	91.3%	89.9%
Rango dispersión diferencia relativa coste total por ejemplar	-	-5.1± 33.2%	-9.8± 21.5%	-10.5± 22.2%	-9.8± 21.7%	-10.5± 22.3%	-11.2± 22.2%

Tabla 8.4: Resultados obtenidos a raíz del estudio del cóctel de heurísticas entre pares de procedimientos.

8.2. Información resultante de la programación de los trabajos

En este apartado se enumera el conjunto de resultados o información que obtendría el usuario al resolver un caso de programación de operaciones con los procedimientos diseñados en este trabajo. Los resultados obtenidos de la programación de operaciones son los siguientes:

- Máquina i_{jh} en que se realiza la operación O_{jh} del trabajo j .
- Instante t_{jh} en el que se inicia la operación O_{jh} del trabajo j .
- Tiempo de proceso de la operación O_{jh} en la máquina m o instante en el que se termina la operación (c_j).
- Disponibilidad final de las máquinas después de haber programado todas las operaciones. Para cada máquina, CF_i será la disponibilidad inicial de la máquina para próximas programaciones.
- Tiempo de retraso o *tardiness* T_j del trabajo j respecto a la fecha de entrega.
- Tiempo de adelanto o *earliness* E_j del trabajo j respecto a la fecha de entrega.
- Coste total de la programación de las operaciones de los trabajos y coste individualizado para cada trabajo.

En la Tabla 8.5 se puede observar un extracto de parte de la información que conformaría el informe de resultados de la programación. Hay que tener en cuenta que los instantes de inicio y fin son incluidos a la hora de contabilizar el tiempo de proceso. Con estos datos, el usuario es capaz de obtener un diagrama de la programación de las operaciones de los trabajos como el mostrado en el Gráfico 8.1. Destacan los saltos entre operaciones que deja el procedimiento utilizado, los cuales pueden producirse por cuatro motivos principales:

1. Debido a la no disponibilidad de máquinas hasta intervalos posteriores.
2. A un retraso del inicio, respecto a la anterior operación, para poder ser programada en una máquina más eficiente y así reducir el coste energético.
3. A un adelanto o retraso del inicio de la operación para poderse ejecutar durante el periodo de hora valle, lo que supone un menor coste eléctrico.
4. Para intentar ajustar el tiempo de finalización del trabajo a la fecha de entrega del mismo.

Trabajo	Operación	Máquina	Instante inicio	Instante de fin	Tiempo proceso
3	1	1	139	165	27
	2	7	300	313	14
	3	2	314	329	16
	4	9	330	353	24
	5	5	595	621	27
5	1	1	504	534	31
	2	2	535	558	24
	3	2	599	646	48
	4	10	647	657	11
	5	10	658	678	21
	6	1	679	684	6
	7	6	685	697	13

Tabla 8.5: Extracto de los resultados que proporciona los procedimientos diseñados al usuario.

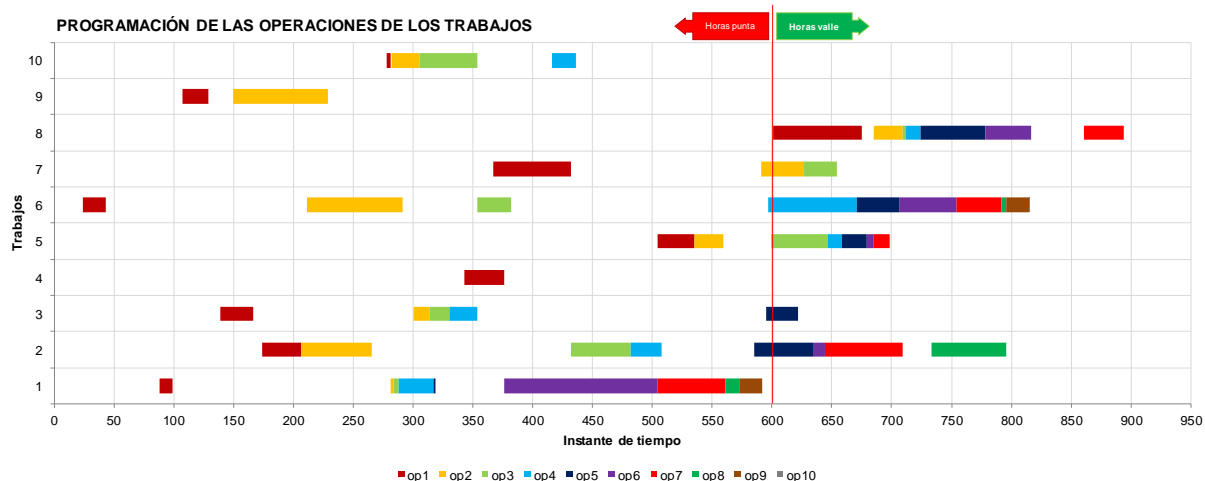


Gráfico 8.1: Diagrama de la programación de operaciones de los trabajos para un ejemplar de 10 trabajos.

8.3. Beneficios logrados

Con la aplicación del procedimiento recomendado en este Trabajo de Fin de Máster (GRASPSA-HeufJSP), el usuario final obtendrá un conjunto de beneficios o ventajas. A continuación, se define el listado de beneficios principales:

- Programación de mayor calidad. Las operaciones de los trabajos son asignadas a las máquinas y programadas de manera que se reduzca el valor del índice de eficiencia de la solución final. Comparativamente con el procedimiento de referencia (Corominas *et al.* 2015), se obtiene una mayor calidad en la mayoría de los tipos de ejemplares.
- Reducción del coste energético. Uno de los objetivos del índice de eficiencia es la

minimización del coste energético, de manera que las soluciones alcanzadas logran reducir el coste energético necesario para producir un trabajo determinado.

- Sistema de producción *just-in-time*. La minimización de los conceptos *earliness* y *tardiness* favorecen a que los trabajos finalicen en la fecha de entrega o muy cerca de ella.
- Reducción de stock de producto acabado: El sistema productivo *just-in-time* tiene el objetivo de producir para el momento preciso y reducir los costes que no aportan valor añadido al producto final. El almacenamiento de producto acabado supone un aumento del coste asociado a éste y un activo material parado para la empresa. Por lo tanto, la reducción del *earliness* va a favor de una reducción de este tipo de stock.
- Evitar las penalizaciones por retraso en la entrega del producto acabado. Muchas empresas negocian con su suministradora la compra de ciertos productos urgentes o vitales para su producción con cláusulas de penalización. Estas cláusulas son aplicables cuando no se cumple con la fecha de entrega pactada. Además, los retrasos sostenidos pueden suponer una mala imagen comercial para la empresa, dificultando así su actividad futura.
- Optimización de recursos productivos. La programación obtenida con la aplicación del procedimiento recomendado permite al usuario tener un mayor control de sus recursos (maquinaria), controlar el mantenimiento de los mismos, las cargas de las máquinas, valorar qué máquinas y de qué tipo son más utilizadas, etc.
- Robustez de la calidad que proporciona el procedimiento delante la variedad de ejemplares del problema *flexible Job-Shop*. Los buenos resultados obtenidos con GRASPSA-HeufJSP, junto con la variedad de los ejemplares testeados en el experimento computacional (1296 combinaciones), convierten a este procedimiento en uno muy robusto delante la variedad de ejemplares a los cuales se puede enfrentar en la realidad.

9. Planificación y costes

La ejecución de todo este Trabajo de Fin de Máster sobre nuevos procedimientos de resolución del problema *flexible Job-Shop* con un sistema productivo *just-in-time* y teniendo en cuenta la discriminación horaria del coste energético, se basa en una planificación ordenada de un conjunto de actividades a ser realizadas. Conjuntamente con la planificación, todo trabajo es valorado por el coste para tener los recursos necesarios y así poder hacer todas las acciones definidas en el plan del trabajo. En los siguientes subapartados se define la planificación que se ha llevado a cabo en este trabajo y una descripción de los costes que ha supuesto el mismo.

9.1. Planificación del trabajo

La planificación de este trabajo se divide en 6 grandes fases. Cada fase está compuesta por varias actividades. El orden con el cual se presentan las fases no es igual al orden de ejecución, sino que las actividades de una misma fase o de fases distintas pueden ejecutarse en paralelo o de manera intercalada (ver Gráfico 9.1). En total, el trabajo ha tenido una duración de 7 meses.

- Fase 1. Análisis preliminar del problema, definición del problema a resolver y estudio del procedimiento de referencia.
- Fase 2. Proceso de investigación y búsqueda de información. Esta fase consiste en encontrar en la literatura distintos métodos de resolución para el problema *flexible Job-Shop*. El objetivo es conocer como se ha resuelto el problema y si la variante estudiada ha sido ya resuelta por otros autores.
- Fase 3. Definición de las especificaciones del problema a resolver, planteamiento de los procedimientos y diseño de los mismos. Esta fase incluye el diseño de cada uno de los procedimientos propuestos en este trabajo, así como su programación y preparación para ser ejecutados.
- Fase 4. Realización de un extenso experimento computacional que sirva para valorar la calidad y funcionalidad de cada uno de los procedimientos diseñados en la fase 3.
- Fase 5. Análisis de los resultados obtenidos en el experimento computacional de la fase 4 y obtención de conclusiones de los mismos.
- Fase 6. Redacción de la documentación relacionada con el Trabajo de Fin de Máster.

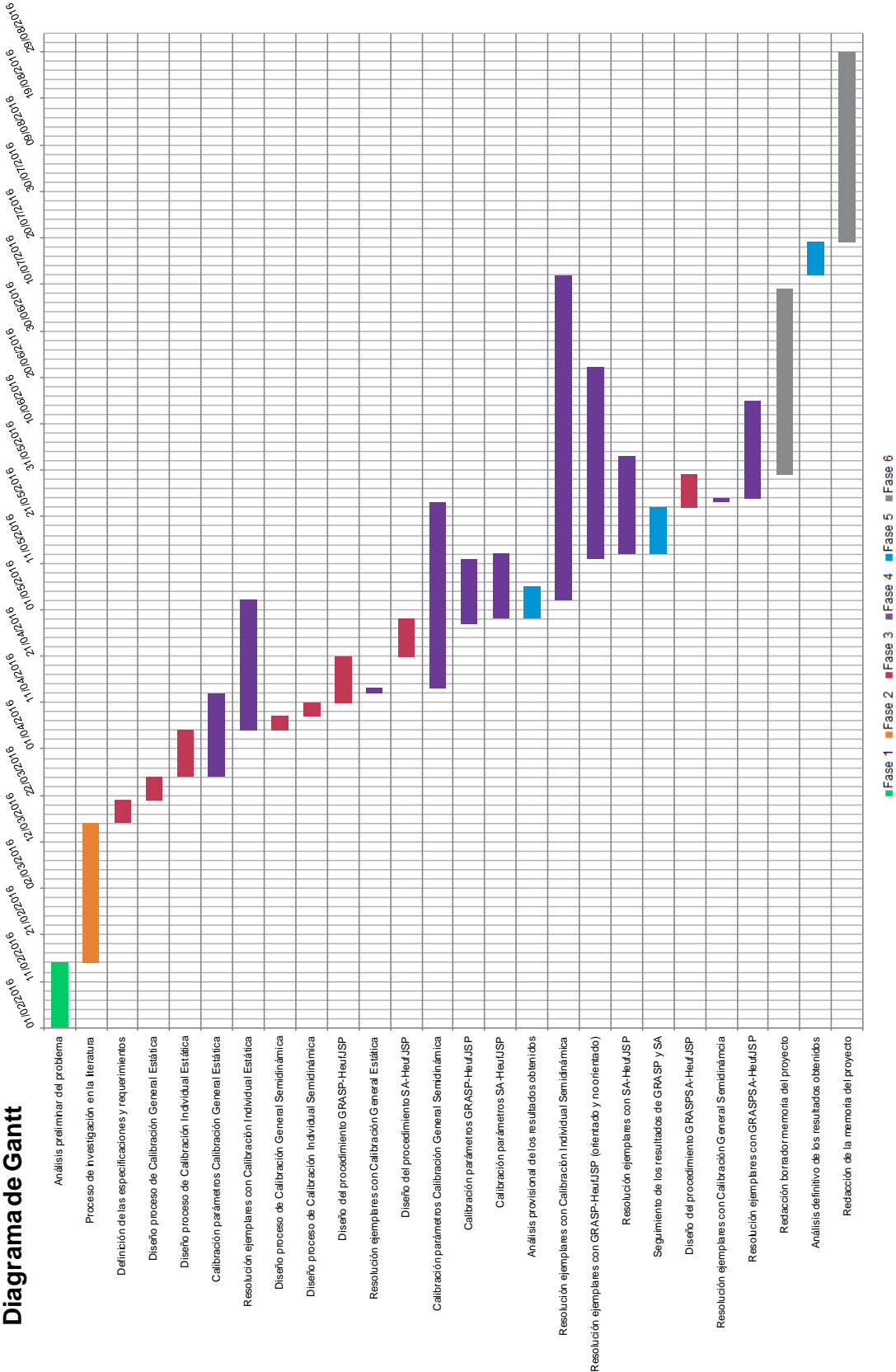


Gráfico 9.1: Diagrama de Gantt de la planificación del presente Trabajo de Fin de Máster.

9.2. Análisis de costes

El análisis de costes del presente trabajo es representado en la Tabla 9.4. Se realiza una división en cinco partidas: partida de costes de personal, tanto del ingeniero diseñador como del director del mismo trabajo; costes de suministro de energía eléctrica; partida de equipamiento informático; partida de mobiliario y partida de material de oficina. Los costes que se presentan tienen en cuenta la previa disponibilidad de una zona o local dónde poder ejecutar el trabajo, de manera que no están incluidos. En las partidas de equipamiento informático y mobiliario se considera una amortización a 4 y 5 años respectivamente, ya que su contenido será usado en posteriores trabajos. La Tabla 9.1, Tabla 9.2 y Tabla 9.3 muestran una segregación en detalle de algunas de las partidas definidas.

PARTIDA DE COSTES DE PERSONAL				
CONCEPTO	CANTIDAD	UNIDAD	COSTE UNITARIO	COSTE TOTAL
Coste ingeniero superior industrial	420	horas	30.00 €	12600.00 €
Coste ingeniero director trabajo	40	horas	60.00 €	2400.00 €
TOTAL PARTIDA DE PERSONAL				15000.00 €

Tabla 9.1: Costes relacionados con la partida de personal.

PARTIDA DE EQUIPAMIENTO INFORMÁTICO				
CONCEPTO	CANTIDAD	UNIDAD	COSTE UNITARIO	COSTE TOTAL
Ordenador de mesa	4	unidad	399.00 €	1596.00 €
Licencia Sistema Operativo Windows 7	4	unidad	142.00 €	568.00 €
Monitor para ordenador	1	unidad	105.00 €	105.00 €
Switch KVM (4 entradas)	1	unidad	20.60 €	20.60 €
Pack ratón y teclado	1	unidad	18.20 €	18.20 €
TOTAL PARTIDA DE EQUIPAMIENTO INFORMÁTICO				2307.80 €
Amortización anual partida equipamiento informático (4 años)				576.95 €
AMORTIZACIÓN DE 7 MESES PARTIDA DE EQUIPAMIENTO INFORMÁTICO				336.55 €

Tabla 9.2: Costes relacionados con la partida de equipamiento informático (amortizable a 4 años).

PARTIDA DE MOBILIARIO	
CONCEPTO	COSTE TOTAL
Mobiliario de oficina	400.00 €
TOTAL PARTIDA DE MOBILIARIO	400.00 €
Amortización anual partida de mobiliario (5 años)	80.00 €
AMORTIZACIÓN DE 7 MESES PARTIDA DE MOBILIARIO	46.67 €

Tabla 9.3: Costes relacionados con la partida de mobiliario (amortizable a 5 años).

El resumen del análisis de costes del presente Trabajo de Fin de Máster es el mostrado en la Tabla 9.4:

CONCEPTO	COSTE TOTAL
Partida 1: Personal	15000.00 €
Partida 2: Suministro de energía eléctrica	300.00 €
Partida 3: Equipamiento informático	336.55 €
Partida 4: Mobiliario	46.67 €
Partida 5: Material de oficina	50.00 €
COSTE TOTAL PARTIDAS	15733.22 €
Margen beneficio (10%)	1573.32 €
SUBTOTAL COSTE	17306.54 €
IVA (21%)	3634.37 €
TOTAL COSTE	20940.91 €

Tabla 9.4: Resumen de los costes totales del presente Trabajo de Fin de Máster.

El coste total del Trabajo de Fin de Máster es de **VEINTE MIL NOVECIENTOS CUARENTA EUROS CON NOVENTA Y UN CÉNTIMOS.**

10. Impacto medioambiental

Un sistema energético mundial compatible con un desarrollo sostenible tiene que abordar cuestiones sobre el impacto medioambiental, las emisiones de gases del efecto invernadero, preservación de los recursos, la vulnerabilidad de interrupción del suministro o la equidad en el acceso a servicios de energía asequibles.

En realidad, el sistema energético mundial está aún lejos de ser sostenible. Dejando al margen las diferencias entre países desarrollados, en vías de desarrollo y del tercer mundo, las cuales afectan a la capacidad de suministro y de acceso a la energía, uno de los principales problemas mundiales es la permanente dependencia de los combustibles fósiles. Este tipo de combustibles contribuyen a la generación de CO_2 al ser combustionados y, en definitiva, aumentan el efecto invernadero. Alrededor del 80% de la energía mundial consumida se fundamenta en los combustibles fósiles (petróleo, carbón y gas natural). La solución a estos problemas es eliminar la dependencia a los combustibles fósiles y modificar el sistema energético dirigiéndolo hacia las energías renovables.

Desde el punto de vista de generación eléctrica, el porcentaje que proviene de los combustibles fósiles baja a un 68% a nivel mundial, según datos de 2013. No obstante, continúa siendo un porcentaje demasiado alto. Los países no están apostando lo suficientemente fuerte con las energías renovables, ya que son tecnologías que necesitan de una gran inversión y el precio unitario de la energía eléctrica continúa siendo más competitivo en las centrales de energía térmica.

Dentro de este contexto, se intuye que el mundo no está concienciado y puesto a dejar los combustibles fósiles a corto plazo, con lo que otro método que se está empleando y, que a priori necesita de una menor inversión, es potenciar la eficiencia energética. En esta dirección van los procedimientos propuestos para resolver el problema *flexible Job-Shop*. La aplicación de estos procedimientos a una industria permitirá conseguir un ahorro del consumo de energía eléctrica.

Parte del objetivo real de los procedimientos presentados es la minimización del coste de la energía eléctrica, que programa las operaciones de los trabajos teniendo en cuenta la discriminación horaria del precio de la energía. De forma indirecta, hay una alta probabilidad de programar las operaciones en las máquinas de menor consumo y priorizar la programación de las operaciones en las horas valle.

La discriminación horaria es un tipo de tarifa muy habitual, con la cual las industrias pueden ahorrarse grandes cuantías de dinero si saben aprovecharla correctamente. Las compañías suministradoras ofrecen el precio de la energía eléctrica variable en función del momento del

día en que se consuma. En general, se suele diferenciar entre horas valle y horas punta. Estas últimas corresponden a las horas de mayor consumo y en que se necesita utilizar métodos de producción eléctrica de rápida reacción, como por ejemplo las plantas de ciclo combinado. Éstas tienen unos costes de generación eléctrica muy elevados que provocan que el coste en horas punta sea mayor. En el bando contrario están las horas valle, con un coste de la electricidad menor, ya que son las horas con menor demanda. Hay ocasiones en que se provoca el consumo de electricidad mediante centrales de bombeo. La razón está en que el sistema de suministro tiene que estar equilibrado entre la generación y el consumo. Programando las operaciones de los trabajos en horas valle se consigue un menor coste unitario de la electricidad y, además, se favorece al equilibrio energético del suministro.

En resumen, los procedimientos planteados fomentan la eficiencia energética de las industrias que tengan un sistema productivo orientado a procesos y se asemejen al problema *flexible Job-Shop*. El listado de ventajas que se consiguen:

1. Alta probabilidad de asignación de las operaciones en las máquinas que menos consumen.
2. Reducción del consumo de energía eléctrica necesaria para producir cada trabajo.
3. Dar prioridad a la programación de operaciones en las horas valle.
4. Disminución del coste energético de la producción, lo que supone una ventaja competitiva para la empresa, ya que pueden bajar los precios de sus productos manteniendo el margen de beneficio.

Conclusiones

En el presente Trabajo de Fin de Máster, se ha resuelto el problema de programación de operaciones *flexible Job-Shop* considerando un tipo de producción *just-in-time* y la discriminación horaria del coste de la energía eléctrica. El objetivo marcado al inicio del trabajo ha sido el diseño de algún procedimiento que permita resolver el problema y, con el cual, se logre una mayor calidad de las soluciones obtenidas en el procedimiento de Corominas *et al.* (2015). Además, se ha fijado el requerimiento que el tiempo máximo por ejemplar no puede superar la hora. Al ser un problema de minimización, la mejor calidad se obtiene cuándo el valor del índice de eficiencia es menor, siendo el índice compartido con el procedimiento de referencia y formado por: los costes de la energía eléctrica necesaria para procesar los trabajos y los costes de adelanto (*earliness*) y retraso (*tardiness*) de los trabajos respecto a la fecha de entrega.

Se han propuesto siete procedimientos distintos en que cada uno tiene sus ventajas e inconvenientes. Los procesos de Calibración General permiten obtener soluciones de manera muy rápida, pero proporcionan una mala calidad de las soluciones. Lo mismo sucede con los dos procedimientos de GRASP-HeufJSP. En cambio, el proceso de Calibración Individual semidinámica es capaz de proporcionar soluciones de alta calidad, pero en el 13% de los ejemplares supera el tiempo límite. Descartando estos procedimientos, se han seleccionado los tres procedimientos con mejor calidad y que cumplen con el requisito de tiempo: Calibración Individual estática con IP^{Cle_3} , SA-HeufJSP y GRASPSA-HeufJSP con azar no orientado. Con el análisis de distintos conceptos, se llega a la conclusión que el procedimiento GRASPSA-HeufJSP es el que mejor comportamiento general tiene. Esto se ratifica con el análisis del mejor procedimiento para cada rango del ratio n/m , en el cual la mayor parte está monopolizada por el procedimiento GRASPSA-HeufJSP. Además, se ha hecho un estudio de cóctel de heurísticas combinando parejas de procedimientos metaheurísticos. El resultado de este estudio es que la pareja SA-HeufJSP + GRASPSA-HeufJSP es la mejor de todas, con una mejora del coste total promedio del 37.5% respecto al procedimiento de referencia (Corominas *et al.* 2015).

La utilización del procedimiento recomendado en este trabajo, GRASPSA-HeufJSP, permitirá al usuario lograr numerosas ventajas: soluciones de mayor calidad, reducción del coste y consumo de energía eléctrica, producción *just-in-time*, reducción del stock de producto acabado y la eliminación o reducción de las penalizaciones por entrega del producto con retraso, entre otras. Todos estos beneficios proporcionan una gran ventaja competitiva a la empresa. La duración total de este Trabajo de Fin de Máster ha sido de 7 meses y su coste se valora en unos 20940.91 € (IVA incluido).

Agradecimientos

La realización y redacción de este Trabajo de Fin de Máster ha requerido de mucho tiempo, esfuerzo y dedicación. Han sido 7 meses muy largos que se han resumido en la presente memoria y que no hubiera sido posible sin la ayuda, en diferentes sentidos, de un grupo de personas que en este apartado querría agradecer.

Desde el punto de vista académico, técnico y teórico, agradecer la dedicación, ayuda y consejos proporcionados tanto por parte del tutor del proyecto, Rafael Pastor Moreno, como del profesor Alberto García-Villoria. Por otro lado, desde el punto de vista personal, agradecer el apoyo moral y motivación que me ha proporcionado mi familia y pareja. Sin ellos no hubiera sido posible. Para finalizar, querría dedicar este trabajo a una persona muy especial en mi vida, un referente como persona. Sé que desde allí donde estés, me habrás ayudado y mucho. Muchas gracias a todos.

Bibliografía

Referencias bibliográficas

- ADENSO-DÍAZ, B. y LAGUNA, M. *CALIBRA V1.0 Manual de usuario*. [Formato .doc]. [s.l.]: Universidad de Oviedo, 2002. [Fecha de consulta: junio de 2016]. Disponible en: http://coruxa.epsig.uniovi.es/~adenso/file_d.html
- ADENSO-DÍAZ, B. y LAGUNA, M. Fine-Tuning of Algorithms Using Fractional Experimental Designs and Local Search. *Operations Research*, 2006, vol. 54, nº 1, pp.99-114.
- BAGHERI, A., ZANDIEH, M., MAHDAVI, I. y YAZDANI, M. An Artificial Immune Algorithm for the flexible Job-Shop Scheduling Problem. *Future Generation Computer Systems*, 2010, vol. 26, nº 4, pp. 533-541.
- BAGHERI, A. y ZANDIEH, M. Bi-criteria flexible job-shop scheduling with sequence-dependent setup times – Variable neighborhood search approach. *Journal of Manufacturing Systems*, 2011, vol. 30, nº 1, pp. 8-15.
- BIRGIN, E., FEOFILOFF, P., FERNANDES, C., DE MELO, E., OSHIRO, M. y RONCONI, D. A MILP model for an extended version of the flexible Job Shop Problem. *Optimization Letters*, 2013, vol. 8, nº 4, pp. 1417-1431.
- BRANDIMARTE, P. Routing and scheduling in a flexible job shop by tabu search. *Annals of Operations Research*, 1993, vol. 41, nº 3, pp. 157-183.
- BRUCKER, P. y SCHLIE, R. Job-shop scheduling with multi-purpose machines. *Computing*, 1990, vol. 45, nº 4, pp. 369-375.
- CALLEJA, G. y PASTOR, R. A dispatching algorithm for flexible job-shop scheduling with transfer batches: an industrial application. *Production Planning & Control*, 2013, vol. 25, nº 2, pp. 93-109.
- CHEN, J., CHEN, K., WU, J. y CHEN, C. A study of the flexible job shop scheduling problem with parallel machines and reentrant process. *The International Journal of Advanced Manufacturing Technology*, 2007, vol. 39, nº 3-4, pp. 344-354.
- CHIANG, T. y LIN, H. A simple and effective evolutionary algorithm for multiobjective flexible Job Shop Scheduling. *International Journal of Production Economics*, 2013,

vol.141, nº 1, pp. 87-98.

- CONWAY, R. W., MAXWELL, W. L., y MILLER, L. W. *Theory of Scheduling*. 1ª edición. Nueva York: Dover Publications, Inc., 2003.
- COROMINAS, A. y PASTOR, R. Designing greedy algorithms for the flow-shop problem by means of Empirically Adjusted Greedy Heuristics (EAGH). *Journal of the Operational Research Society*, 2011, vol. 62, nº 9, pp. 1704-1710.
- COROMINAS, A., GARCÍA-VILLORIA, A. y PASTOR, R. Cóctel de heurísticas para resolver problemas difíciles. *DYNA*, 2012, vol. 87, nº 3, pp. 275-278.
- COROMINAS, A., GARCÍA-VILLORIA, A. y PASTOR, R. Improving parametric Clarke and Wright algorithms by means of iterative Empirically Adjusted Greedy Heuristics. *Statistics and Operations Research Transactions*, 2014, vol. 38, pp. 3-12.
- COROMINAS, A., GARCÍA-VILLORIA, A., GONZÁLEZ, N. A. y PASTOR, R. *A multistage-graph based procedure for solving a just-in-time flexible job-shop scheduling problem with time dependent costs*. ETSEIB - Universitat Politècnica de Catalunya, Barcelona, 2015.
- DAUZÈRE-PÉRÈS, S. y PAULLI, J. An integrated approach for modeling and solving the general multiprocessor Job-Shop Scheduling Problem using Tabu Search. *Annals of Operations Research*, 1997, vol. 70, pp. 281-306.
- DE GIOVANNI, L. y PEZZELLA, F. An Improved Genetic Algorithm for the Distributed and flexible Job-Shop Scheduling Problem. *European Journal of Operational Research*, 2010, vol. 200, nº 2, pp. 395-408.
- FATTAHI, P., SAIDI MEHRABAD, M. y JOLAI, F. Mathematical modeling and heuristic approaches to flexible job shop scheduling problems. *J Intell Manuf*, 2007, vol. 18, nº 3, pp.331-342.
- FATTAHI, P., JOLAI, F. y ARKAT, J. Flexible Job Shop Scheduling with overlapping in operations. *Applied Mathematical Modelling*, 2009, vol. 33, nº 7, pp. 3076-3087.
- FEO, T.A. y RESENDE, M.G.C. A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 1989, vol. 8, pp. 67-81.
- FIRAS, A. y DENIS, G. A multi-objective genetic method minimizing tardiness and energy consumption during idle times. *IFAC-PapersOnLine*, 2015, vol. 48, nº 3, pp.1216-1223.

- GEN, M., GAO, J. y LIN, L. Multistage-Based Genetic Algorithm for flexible Job-Shop Scheduling Problem. *Intelligent and Evolutionary Systems*, 2009, vol. 187, pp. 183-196.
- GOMES, M., BARBOSA-POVOA, A. y NOVAIS, A. Scheduling of Job-Shop, make-to-order industries with recirculation and assembly: a MILP approach. *2008 IEEE International Engineering Management Conference*, 2008, pp. 1-6.
- GONZÁLEZ, M., VELA, C. y VARELA, R. Scatter Search with Path Relinking for the flexible Job Shop Scheduling Problem. *European Journal of Operational Research*, 2015, vol. 245, nº 1, pp. 35-45.
- HE, Y., LI, Y., WU, T. y Sutherland, J. An energy-responsive optimization method for machine tool selection and operation sequence in flexible machining job shops. *Journal of Cleaner Production*, 2015, vol. 87, pp. 245-254.
- HUANG, R., YANG, C. y CHENG, W. Flexible job shop scheduling with due window – a two-pheromone ant colony approach. *International Journal of Production Economics*, 2013, vol. 141, nº 2, pp. 685-697.
- HURINK, J., JURISCH, B. y THOLE, M. Tabu Search for the Job-Shop Scheduling Problem with multi-purpose machines. *OR Spektrum*, 1994, vol. 15, nº 4, pp. 205-215.
- ISHIKAWA, S., KUBOTA, R. y HORIO, K. Effective hierarchical optimization by a hierarchical multi-space competitive Genetic Algorithm for the flexible Job-Shop Scheduling. *Expert Systems with Applications*, 2015, vol. 42, nº 24, pp. 9434-9440.
- JIANG, Z., ZUO, L. y E, M. Study on multi-objective flexible job-shop scheduling problem considering energy consumption. *Journal of Industrial Engineering and Management*, 2014, vol. 7, nº 3, pp. 589-604.
- KIRKPATRICK, S., GELATT, C.D. y VECCHI, M.P. Optimization by Simulated Annealing. *Science*, 1983, vol. 220, pp. 671-680.
- LARROSA, C. y MESEGUER, P. Generic CSP techniques for the job-shop problem. *Tasks and Methods in Applied Artificial Intelligence*, 1998, pp. 46-55.
- LI, L. y HUO, J. Multi-Objective Flexible Job-Shop Scheduling Problem in Steel Tubes Production. *Systems Engineering – Theory & Practice*, 2009, vol. 29, nº 8, pp. 117-126.
- LI, J., PAN, Q. y XIE, S. An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems. *Applied Mathematics and Computation*, 2012, vol. 218, nº 18, pp. 9353-9371.

- LI, X. y GAO, L. An effective hybrid Genetic Algorithm and Tabu Search for flexible Job Shop Scheduling Problem. *International Journal of Production Economics*, 2016, vol.174, pp. 93-110.
- LIU, Y., DONG, H., LOHSE, N., PETROVIC, S. y GINDY, N. An investigation into minimizing total energy consumption and total weighted tardiness in job shops. *Journal of Cleaner Production*, 2014, vol. 65, pp. 87-96.
- LIU, Y. y TIWARI, A. An Investigation into Minimising Total Energy Consumption and Total Completion Time in a Flexible Job Shop for Recycling Carbon Fiber Reinforced Polymer. *Procedia CIRP*, 2015, vol. 29, pp. 722-727.
- MASTROLILLI, M. y GAMBARDELLA, L. Effective neighborhood functions for the flexible Job-Shop Problem. *Journal of Scheduling*, 2000, vol. 3, nº 1, pp. 3-20.
- MOUZON, G. *Operational methods and models for minimization of energy consumption in a manufacturing environment*. Tesis doctoral. Wichita State University. Whicita, Kansas (Estados Unidos), 2008.
- MUTH, J. y THOMPSON, G. *Industrial scheduling*. Englewood Cliffs, Nueva Jersey (Estados Unidos): Prentice-Hall, 1963.
- NAJID, N. M., DAUZÈRE-PÉRÈS, S. y ZAIDAT, A. A modified Simulated Annealing method for flexible Job Shop Scheduling Problem. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2002, vol. 5, pp. 89-94.
- NELDER, J. y MEAD, R. A Simplex Method for Function Minimization. *The Computer Journal*, 1965, vol. 7, nº 4, pp. 308-313.
- ROSHANAIEI, V., AZAB, A. y ELMARAGHY, H. Mathematical modelling and a meta-heuristic for flexible Job Shop Scheduling. *International Journal of Production Research*, 2013, vol. 51, nº 20, pp. 6247-6274.
- ROSSI, A. y DINI, G. Flexible Job-Shop Scheduling with routing flexibility and separable setup times using Ant Colony Optimisation method. *Robotics and Computer-Integrated Manufacturing*, 2007, vol. 23, nº 5, pp. 503-516.
- SAIDI-MEHRABAD, M. y FATTAHI, P. Flexible Job Shop Scheduling with Tabu Search algorithms. *The International Journal of Advanced Manufacturing Technology*, 2006, vol.32, nº 5-6, pp. 563-570.
- SCRICH, C., ARMENTANO, V. y LAGUNA, M. Tardiness minimization in a flexible job

- shop: A tabu search approach. *Journal of Intelligent Manufacturing*, 2004, vol. 15, nº 1, pp. 103-115.
- SHAHSAVARI-POUR, N. y GHASEMISHABANKAREH, B. A novel hybrid meta-heuristic algorithm for solving multi objective flexible Job Shop Scheduling. *Journal of Manufacturing Systems*, 2013, vol. 32, nº 4, pp. 771-780.
 - SINGH, M. y MAHAPATRA, S. A quantum behaved particle swarm optimization for flexible Job Shop scheduling. *Computers & Industrial Engineering*, 2016, vol. 93, pp. 36-44.
 - SONG, W., ZHANG, C., LIN, W. y SHAO, X. Flexible Job-Shop Scheduling Problem with Maintenance Activities Considering Energy Consumption. *AMM*, 2014, vol. 521, pp.707-713.
 - TANEV, I., UOZUMI, T. y MOROTOME, Y. Hybrid evolutionary algorithm-based real-world flexible job shop scheduling problem: application service provider approach. *Applied Soft Computing*, 2004, vol. 5, nº 1, pp. 87-100.
 - TAY, J. y HO, N. Evolving dispatching rules using genetic programming for solving multi-objective flexible job-shop problems. *Computers & Industrial Engineering*, 2008, vol. 54, nº 3, pp. 453-473.
 - THAMMANO, A. y PHU-ANG, A. A Hybrid Artificial Bee Colony Algorithm with Local Search for Flexible Job-Shop Scheduling Problem. *Procedia Computer Science*, 2013, vol. 20, pp.96-101.
 - THÖRNBLAD, K., STRÖMBERG, A., PATRIKSSON, M. y ALMGREN, T. Scheduling optimization of a real flexible Job Shop including fixture availability and preventive maintenance. *EJIE*, 2015, vol. 9, nº 1, pp. 126.
 - VAGHEFINEZHAD, S. y WONG, K.Y. A Genetic Algorithm approach for solving a flexible Job-Shop Scheduling Problem. *IJCSI – International Journal of Computer Science Issues*, 2012, vol. 9, nº 3, pp. 85-90.
 - VILCOT, G. y BILLAUT, J. A Tabu Search algorithm for solving a multicriteria flexible Job Shop Scheduling Problem. *International Journal of Production Research*, 2011, vol.49, nº 23, pp. 6963-6980.
 - WU, Z. y WENG, M. Multiagent Scheduling Method With Earliness and Tardiness Objectives in Flexible Job Shops. *IEEE Transactions on Systems, Man and Cybernetics Part B*, 2005, vol. 35, nº 2, pp. 293-301.

- XIA, W. y WU, Z. An effective hybrid optimization approach for multiobjective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, 2005, vol. 48, nº 2, pp.409-425.
- ZARROUK, R., ETTOUIL, M., BENNOUR, I. y JEMAI, A. Towards an Embedded Distributed Implementations of PSO Solutions for the Flexible Job Shop Problem. *Procedia Computer Science*, 2015, vol. 73, pp. 146-153.

Bibliografía complementaria

- COROMINAS, A., GARCÍA-VILLORIA, A. y PASTOR, R. Fine-tuning a parametric Clarke and Wright heuristic by means of EAGH (Empirically Adjusted Greedy Heuristics). *Journal of the Operational Research Society*, 2009, vol. 61, nº 8, pp. 1309-1314.
- COROMINAS, A., GARCÍA-VILLORIA, A. y PASTOR, R. Technical Note: Relating to the Parameter Values Given by Nelder and Mead in their Algorithm. *The Computer Journal*, 2013, vol. 58, nº 1, pp.157-159.
- Docs.oracle.com. *Overview (Java Platform SE 7)*. [en línea]. [Fecha de consulta: abril de 2016]. Disponible en: <https://docs.oracle.com/javase/7/docs/api/overview-summary.html>
- lea.org. *Climate change*. [en línea]. [Fecha de consulta: 8 de agosto de 2016]. Disponible en: <https://www.iea.org/topics/climatechange/>
- PASTOR, R. *Transparencias: Métodos Cuantitativos de Organización Industrial II. Curso 2015-2016_1S*. ETSEIB - Universitat Politècnica de Catalunya, Barcelona, 2015-2016.

Anexos

Anexo 1: Grafo de múltiples etapas (HeufJSP)

La heurística HeufJSP fue diseñada por Corominas *et al.* (2015) y se emplea en su algoritmo, HeufJSP(ζ), para resolver el problema *flexible Job-Shop*. Dos son las fases que conforman dicha heurística:

1. Obtención de una secuencia ordenada de trabajos.
2. Programación óptima de las operaciones de cada trabajo.

En este anexo se describe con detalle la segunda fase de la heurística HeufJSP, la cual es la responsable que se considere como enfoque integrado esta heurística. Para realizar la programación de las operaciones, se aplica un grafo de múltiples etapas distinto para cada trabajo. La programación consiste en calcular el camino mínimo del grafo, siempre cumpliendo con el orden de precedencias de las operaciones y la disponibilidad de las máquinas.

El grafo de múltiples etapas se obtiene mediante dos pasos: construcción del grafo y cálculo del camino mínimo del mismo, los cuales son descritos de forma detallada en los siguientes apartados de este anexo. Cabe destacar que aunque se realice esta división, en la realidad se dan de forma simultánea, desde principio a fin del grafo. Para más información consultar Corominas *et al.* (2015).

Paso 1: Construcción del grafo de múltiples etapas

Para cada trabajo j , se construye un grafo de múltiples etapas con un total de $h_j + 2$ etapas, iniciándose en la etapa 0 y terminando la programación en la etapa $h_j + 1$. Los arcos que van de la etapa $h - 1$ a la etapa h , siendo $h = 1, \dots, h_j$, corresponden a la programación de la operación O_{jh} . Por otro lado, el arco que sale de la etapa h_j hacia la etapa final $h_j + 1$ representa la terminación del trabajo.

Las etapas 1 a h_j pueden contener un nodo o más de uno (v_{ht}). Estos nodos corresponden a los instantes t en que se puede terminar el proceso de las operaciones O_{jh} . Las etapas 0 y $h_j + 1$ son las únicas que contienen un único nodo: α o nodo de inicio del grafo y ω o nodo final del grafo, respectivamente.

Como en todo grafo, los nodos de las diferentes etapas se comunican entre sí mediante arcos. En función de cuales sean las etapas, el coste asociado al arco tendrá un significado u otro. Todos los arcos entre las etapas 0 y h_j hacen referencia a la programación de la operación O_{jh} en la máquina i , teniendo en cuenta que la máquina pertenece al grupo de máquinas capacitadas para realizar esta operación ($i \in M_{jh}$). El coste asociado a estos arcos es la energía necesaria para efectuar la operación (P_{ijh}). Una peculiaridad de la etapa h_j es que todos los nodos están relacionados con la última etapa o nodo ω . El coste de estos arcos hace referencia a la suma del *earliness* y *tardiness* del trabajo j , con respecto a la fecha de entrega o vencimiento d_j .

Podría darse el caso que hubiera dos arcos que salieran de un mismo nodo de la etapa $h < h_j$ hacia la etapa $h+1 \leq h_j$. Los arcos corresponderían a máquinas distintas y uno de ellos se podría eliminar aplicando la regla R1. Se pueden dar dos casos:

- Caso R1a: Si un arco tiene un coste asociado superior al otro arco, se puede eliminar manteniendo el de menor coste.
- Caso R1b: Si los dos arcos tienen el mismo coste, se mantiene el arco que corresponde a la máquina con menor estimación de la carga de trabajo futura (q_{ijh}).

En el caso que aún hubiera empate, se elimina el arco de la máquina que se identifica con el valor más alto de i .

La estimación de la carga de trabajo futura q_{ijh} se calcula como el tiempo de proceso de la operación O_{jh} en la máquina i , más la suma de cocientes de los tiempos de proceso de las operaciones de otros trabajos que podrían ser ejecutadas en la misma máquina entre el número de máquinas en que cada operación podría ser procesada.

Se aplica una segunda regla para definir la dominancia entre arcos que emanan de un mismo nodo, perteneciente a la etapa $h \leq h_j - 1$, y van a diferentes nodos de la etapa siguiente. Esta regla se denomina R2 y se pueden dar hasta 3 ocurrencias distintas:

- Caso R2a: Para los arcos que salen del mismo nodo de la etapa $h < h_j - 1$, si el coste de un arco no es mejor que otro y se dirige a un nodo que representa un intervalo de finalización mayor, se elimina el arco.
- Caso R2b: Para los arcos que salen del mismo nodo de la etapa $h = h_j - 1$ y se dirigen a distintos nodos de la etapa h_j que representan $t \geq d_j$, si el coste de un arco no es mejor a otro y se dirige a un nodo que representa un intervalo de

finalización mayor, se elimina el arco.

- **Caso R2c:** Para los arcos que salen del mismo nodo de la etapa $h = h_j - 1$ y se dirigen a distintos nodos de la etapa h_j que representan $t < d_j$, si el coste de un arco no es mejor a otro y se dirige a un nodo que representa un intervalo de finalización menor, se elimina el arco.

Los arcos que son omitidos por el caso R2a, cuando el coste del arco dominado es mayor que el del arco dominante, R2b y R2c no podrían incumbir al camino mínimo. Cuando los arcos dominado y dominante tienen el mismo coste (caso R2a), el dominado puede ser omitido sin que perjudique al coste del camino mínimo del grafo de múltiples etapas. El método escogido para romper los empates favorece la asignación que permita realizar las operaciones antes.

Paso 2: Cálculo del camino mínimo

Tal y como se ha comentado con anterioridad, el problema *flexible Job-Shop* es resuelto de forma óptima, para cada trabajo j , calculando el camino mínimo del grafo, el cual se realiza de forma paralela a la construcción del mismo (paso 1).

Para ir encontrando el camino mínimo, a cada nodo se le atribuye un coste según la aplicación de las siguientes fórmulas:

$$\kappa_{0\alpha} = 0 \quad (\text{Ec. A1.1})$$

$$\kappa_{ht'} = \min_{\forall t | v_{h-1,t} \in \Gamma_{ht'}^-} (\kappa_{h-1,t} + c_{htt'}) \quad (h = 1, \dots, h_j + 1) \quad (\text{Ec. A1.2})$$

Donde:

$\kappa_{0\alpha} \equiv$ Valor del nodo inicial α

$\kappa_{ht} \equiv$ Valor del nodo v_{ht}

$c_{htt'} \equiv$ Coste del arco que conecta el nodo $v_{h-1,t}$ con el nodo $v_{ht'}$

$\Gamma_{ht'}^- \equiv$ Conjunto de nodos de la etapa $h - 1$ que están conectado con el nodo $v_{ht'}$

Los empates que se pueden llegar a producir se desempatan con una tercera regla, la cual contempla 3 casos:

- Caso R3a (arcos que se dirigen a los nodos de la etapa $h \leq h_j$): Si hay empate entre arcos que tienen el mismo coste y corresponden a la misma máquina, el arco que se mantiene es el que proviene del nodo que tiene un menor tiempo de finalización de la operación previa ($O_{j,h-1}$).
- Caso R3b (arcos que se dirigen al nodo ω): El arco que se mantiene es el que emana del nodo correspondiente al menor tiempo de finalización de la última operación del trabajo (O_{j,h_j}).
- Caso R3c (arcos que se dirigen a los nodos de la etapa $h \leq h_j$): Si el empate es entre arcos que corresponde a distintas máquinas, el arco que se mantiene es el de la máquina con una estimación de la carga de trabajo menor (q_{ijh}). Si aún hubiera empate con algunos arcos, el que se mantendría es el que emana del nodo que tiene un menor tiempo de finalización de la operación previa ($O_{j,h-1}$).

Además, existe una cuarta regla (R4) que hace referencia a los nodos pertenecientes a la misma etapa ($1 \leq h \leq h_j - 1$). Si hay dos nodos en una misma etapa, v_{ht} y $v_{ht'}$, con $\kappa_{ht} \leq \kappa_{ht'}$ y $t < t'$, entonces el nodo v_{ht} domina al nodo $v_{ht'}$, el cual puede ser eliminado ya que no puede pertenecer al camino mínimo del grafo.

Anexo 2: Algoritmos de calibración

Este anexo incluye una descripción más amplia de los tres algoritmos que se han utilizado en los procedimientos presentados para resolver el *flexible Job-Shop Problem*. El objetivo no es hacer una descripción detallada, sino que se trata de definir los conceptos generales para cada algoritmo.

EAGH: Empirically Adjusted Greedy Heuristics

El procedimiento *Empirically Adjusted Greedy Heuristics* (EAGH) fue presentado por Corominas y Pastor (2011). Existen heurísticas, como las *greedy*, que encuentran una solución utilizando un indicador. Un indicador es una función f formada por los atributos del problema y varios parámetros que los ponderan (en el problema planteado el indicador sería el índice de prioridad). En general, los indicadores tienen la finalidad de seleccionar y/u ordenar un conjunto. Potencialmente, existe un conjunto infinito de indicadores F , resultado de las infinitas combinaciones que se pueden generar con los parámetros.

EAGH es un procedimiento que busca el mejor indicador $f \in F$. La definición de mejor depende de la función objetivo o índice de eficiencia a minimizar (FO) y de un conjunto de ejemplares de prueba K . Siendo $X_{k,\Pi}$ la solución del ejemplar $k \in K$ obtenida aplicando la heurística, entonces la función φ queda definida como (Ec. A2.1):

$$\varphi(\Pi) = \sum_{k \in K} FO(X_{k,\Pi}) \quad (\text{Ec. A2.1})$$

Cabe destacar que encontrar el mejor indicador equivale a encontrar los valores de los parámetros Π que minimicen la función φ . Por lo tanto, una clara aplicación del procedimiento EAGH es la calibración de los valores de los parámetros del indicador.

Sólo es posible utilizar un algoritmo de optimización directa, es decir, un algoritmo de optimización que sólo utilice los valores de la función. La razón recae en el hecho que la función φ no tiene ninguna propiedad especial ni reconocible que permita utilizar otra tipología de algoritmo. Un ejemplo de algoritmo de optimización directa es *Nelder & Mead*.

Nelder & Mead

El algoritmo de optimización directa *Nelder & Mead*, también conocido como *Flexible Polyhedron Algorithm*, recibe el nombre de sus dos autores que lo presentaron en el año 1965. Se trata de uno de los algoritmos más usados para resolver problemas de optimización no restringidos.

Nelder & Mead es un algoritmo de optimización directa para minimizar $f(x)$, donde $f: \mathbf{R}^n \rightarrow \mathbf{R}$ es la función objetivo y n la dimensión. Se basa en $n+1$ puntos que son los vértices de un *simplex* de dimensión n : x_1, x_2, \dots, x_{n+1} . Por ejemplo, en el caso de aplicación que se presenta en este trabajo, la dimensión n es igual al número de parámetros a calibrar (no confundir con el número de trabajos). Cada punto tiene asociado n coordenadas, que son los valores que toman los parámetros en ese punto. Para cada punto, la resolución del problema y la aplicación del índice de eficiencia o función objetivo permite asociar un valor al punto que indica su calidad.

El método empieza desde un *simplex* inicial (hipertetraedro) e iterativamente va moviendo los vértices por el espacio n -dimensional, siempre de acuerdo a los valores de la función objetivo para cada punto y hasta alcanzar las condiciones de fin.

A cada iteración de *Nelder & Mead*, los vértices del *simplex* son etiquetados y ordenados según su valor de la función objetivo: $f(x_1) \leq f(x_2) \leq \dots \leq f(x_{n+1})$. En caso de empate, el vértice más antiguo tiene prioridad. Siendo $x_r = \bar{x} + \alpha \cdot (\bar{x} - x_{n+1})$ la reflexión de x_{n+1} , donde \bar{x} es el baricentro de los n mejores vértices ($\bar{x} = \sum_{i=1}^n x_i / n$) y $\alpha > 0$ es un parámetro. Cuatro son los casos que son considerados de acuerdo al valor de $f(x_r)$ (ver Figura A2.1):

1. **Expansión:** Si $f(x_r) < f(x_1)$ se tiene que calcular $x_e = \bar{x} + \gamma \cdot (x_r - \bar{x})$, donde $\gamma > 1$ es un parámetro. Si $f(x_e) < f(x_1)$, reemplazar x_{n+1} por x_e . En caso contrario, reemplazar x_{n+1} por x_r .
2. **Reflexión:** Si $f(x_1) \leq f(x_r) < f(x_n)$, entonces se tiene que reemplazar x_{n+1} por x_r .
3. **Contracción externa:** Si $f(x_n) \leq f(x_r) < f(x_{n+1})$, entonces calcular $x_{oc} = \bar{x} + \beta \cdot (x_r - \bar{x})$, donde $0 < \beta < 1$ es un parámetro. Si $f(x_{oc}) < f(x_r)$, reemplazar x_{n+1} por x_{oc} ; en caso contrario, reemplazar x_{n+1} por x_r y contraer todos los vértices, excepto x_1 , de la siguiente manera: $x_i = x_1 + \delta \cdot (x_i - x_1)$ $i = 2, \dots, n+1$, donde $0 < \delta < 1$ es un parámetro.
4. **Contracción interna:** Si $f(x_{n+1}) \leq f(x_r)$, entonces hay que calcular $x_{ic} = \bar{x} + \beta \cdot (x_{n+1} - \bar{x})$. Si $f(x_{ic}) < f(x_{n+1})$, reemplazar x_{n+1} por x_{ic} ; en caso contrario, encoger todos los vértices, excepto x_1 , igual que en el caso 3.

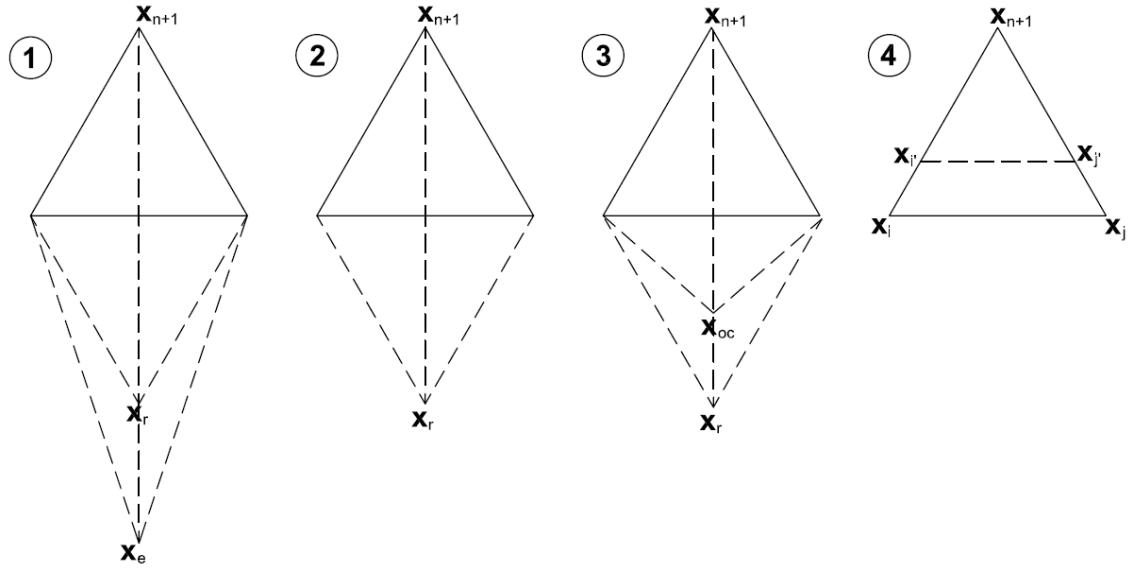


Figura A2.1: Los cuatro casos que se pueden dar en el proceso de *Nelder & Mead*.

Los parámetros α , β , γ y δ son usados en los casos de reflexión, contracción, expansión y encogimiento, respectivamente. Los autores del procedimiento *Nelder & Mead* establecieron el valor de los parámetros como 1, 0.5, 2 y 0.5, respectivamente. Desde entonces, estos valores han sido utilizados en la amplia mayoría de las aplicaciones del algoritmo *Nelder & Mead*.

El algoritmo termina cuando se cumplen las condiciones de fin, $S1$ y $S2$. Por lo tanto, *Nelder & Mead* termina su ejecución cuando las diferencias entre los valores de los vértices son lo suficientemente pequeñas (condición $S1$) y el *simplex* es lo suficientemente pequeño (condición $S2$). Las expresiones de las condiciones son las siguientes:

$$S1 = f(x_{n+1}) - f(x_1) \leq \varepsilon |f(x_1)| \quad (\text{Ec. A2.2})$$

$$S2 = \max(\alpha \cdot \gamma, 1)^{ne} \cdot \delta^{ns} \cdot L \leq \varepsilon \quad (\text{Ec. A2.3})$$

Siendo:

$\varepsilon \equiv$ Parámetro interno del algoritmo ($\varepsilon = 10^{-8}$).

$ne \equiv$ Número de expansiones del procedimiento.

$ns \equiv$ Número de encogimientos del procedimiento.

$L \equiv$ Longitud de los lados del *simplex* inicial, los cuales se asumen como regulares.

CALIBRA

El programa CALIBRA fue desarrollado por Adenso-Díaz y Laguna (2006) y es un procedimiento implementado en software totalmente gratuito (*freeware*). El propósito del programa consiste en ser utilizado en aquellos procedimientos que contengan parámetros a calibrar, de aquí su nombre. Para ello, CALIBRA realiza una exploración del espacio de posibles valores aplicando un diseño de experimentos factorial de tipo Taguchi, más un proceso de búsqueda local.

Dado un algoritmo con un conjunto de parámetros a calibrar, CALIBRA prepara una serie de experimentos basados en la metodología Taguchi. Estos experimentos tienen el objetivo de encontrar el mejor valor para cada uno de los parámetros. El programa está limitado para calibrar cinco parámetros a la vez. En caso que hubiera más, los autores aconsejan fijar los considerados como menos importantes.

CALIBRA se encarga de llamar a ejecutar el algoritmo y le va pasando distintos valores de los parámetros para calcular cada uno de los experimentos. El usuario habrá definido previamente los intervalos, dentro de los cuales estará el valor de cada parámetro (BOUNDS.DAT). Luego, el programa lee un fichero de texto generado por el algoritmo, en el cual se guardan las respuestas o soluciones del experimento (COST.DAT). Estas respuestas son tratadas por el programa y son utilizadas para generar nuevos valores de los parámetros a calibrar (PARAM.DAT). El proceso termina según la condición de fin definida por el usuario. La Figura A2.2 muestra un esquema del funcionamiento de CALIBRA.

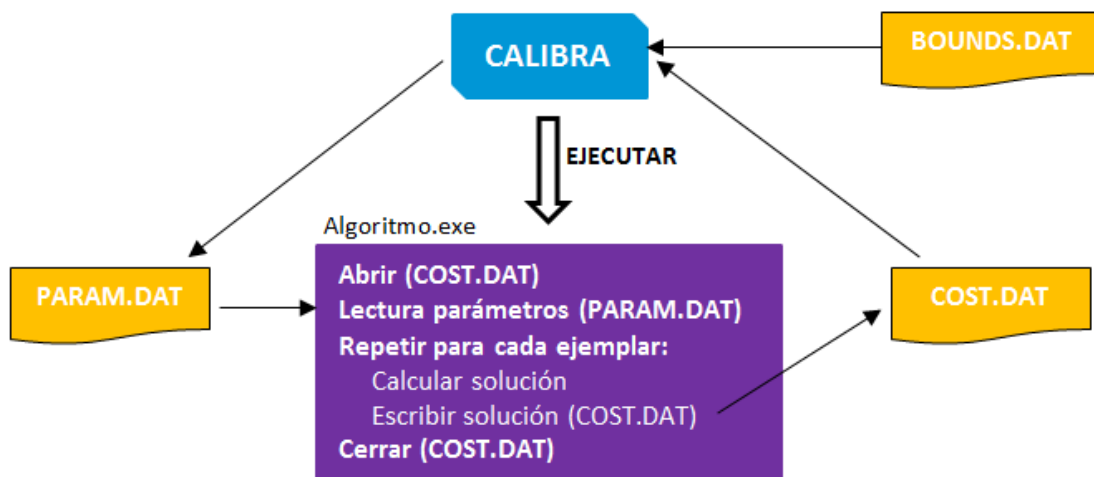


Figura A2.2: Esquema de funcionamiento del programa CALIBRA (Adenso-Díaz y Laguna 2002).

Anexo 3: Tablas de resultados

En este anexo, se muestra un conjunto de tablas de resultados fruto del experimento computacional realizado con los 7 procedimientos diseñados. En las tablas de comparación, el signo negativo significa mejora respecto al procedimiento de referencia (Corominas *et al.* 2015).

Los resultados están desglosados según el ratio n/m (trabajos/máquinas), el cual se cree que es el mejor método para clasificar los ejemplares utilizados en el experimento computacional. Las dos últimas columnas de las tablas, con excepción de la Tabla A3.1, siguen una escala de color, la cual permite una mejor visualización de los resultados. Los tonos rojos son para los valores más pequeños de la columna (peores) y los verdes para los más grandes (mejores). Las dos columnas de una misma tabla son consideradas de forma independiente, por lo que un valor puede tener un color en la primera columna y, en la segunda, tener un color distinto.

En la Tabla A3.1 se presentan los resultados obtenidos en el artículo de referencia, que serán fuente de comparativa para las siguientes tablas de este anexo.

Ratio n/m	Tiempo promedio por ejemplar [s]	Coste total promedio [€]	Coste energía promedio [€]	Coste tardiness y earliness promedio [€]
(0,1]	36.8	313.9	247.1	66.7
(1,2]	100.6	1156.8	602.6	554.1
(2,3]	92.2	3755.2	858.4	2896.8
(3,4]	60.6	9941.3	1112.1	8829.1
(4,5]	37.7	31211.7	1260.8	29950.8
(5,6]	46.4	38034.5	1534.9	36499.7
7	27.8	77692.5	1570.9	76121.6
8	33.1	80679.2	1740.7	78938.5
9	36.2	97511.7	1995.3	95516.3
10	42.4	158385.0	2304.1	156080.9
11	49.3	173604.1	2521.8	171082.3
12	58.5	206072.7	2784.6	203288.2
TOTAL	67.4	15795.5	779.5	15016.0

Tabla A3.1: Resultados promedio del procedimiento HeufJSP(ζ) desglosado por rangos n/m .

Calibración General estática

Ratio n/m	Total ejemplares	Tiempo promedio por ejemplar [s]	Diferencia relativa del tiempo	Coste total promedio [€]	Diferencia relativa del coste total	Coste energía promedio [€]	Diferencia relativa coste energía	Coste tardiness y earliness promedio [€]	Diferencia relativa coste tardiness y earliness	Tardiness promedio	Earliness promedio	Ejemplares mejora coste total	Ejemplares mejora o iguala coste total
(0,1]	1134	2.2	-94.0%	496.5	58.2%	247.2	0.02%	249.3	273.7%	0.7	0.2	0.2%	71.4%
(1,2]	1134	5.8	-94.2%	2031.9	75.7%	603.1	0.07%	1428.9	157.9%	1.4	0.5	2.7%	33.1%
(2,3]	648	5.4	-94.2%	5900.2	57.1%	860.9	0.30%	5039.3	74.0%	3.6	1.4	4.3%	15.9%
(3,4]	324	3.6	-94.1%	13609.9	36.9%	1111.8	-0.03%	12498.0	41.6%	6.6	2.7	5.2%	9.3%
(4,5]	162	2.2	-94.1%	46475.8	48.9%	1262.9	0.16%	45212.9	51.0%	13.6	4.0	7.4%	10.5%
(5,6]	162	2.8	-94.1%	45315.5	19.1%	1541.2	0.41%	43774.3	19.9%	13.8	4.9	13.6%	15.4%
7	54	1.6	-94.3%	121267.1	56.1%	1584.5	0.87%	119682.6	57.2%	26.9	8.2	7.4%	11.1%
8	54	1.9	-94.3%	108631.8	34.6%	1755.1	0.83%	106876.7	35.4%	23.1	10.0	20.4%	20.4%
9	54	2.1	-94.1%	161978.5	66.1%	1998.3	0.15%	159980.1	67.5%	28.6	9.5	14.8%	14.8%
10	54	2.7	-93.7%	203716.5	28.6%	2326.6	0.97%	201389.9	29.0%	31.2	11.3	20.4%	20.4%
11	54	2.8	-94.4%	259302.4	49.4%	2554.1	1.28%	256748.3	50.1%	32.8	10.9	9.3%	9.3%
12	54	3.4	-94.2%	298633.3	44.9%	2812.9	1.02%	295820.4	45.5%	39.9	10.8	11.1%	11.1%
TOTAL	3888	3.9	-94.2%	22700.9	43.7%	782.0	0.32%	21918.9	46.0%	5.4	1.9	4.0%	36.2%

Tabla A3.2: Resultados promedio de Calibración General estática comparada con HeufJSP(ζ).

Calibración General semidinámica

Ratio n/m	Total ejemplares	Tiempo promedio por ejemplar [s]	Diferencia relativa del tiempo	Coste total promedio [€]	Diferencia relativa del coste total	Coste energía promedio [€]	Diferencia relativa coste energía	Coste tardiness y earliness promedio [€]	Diferencia relativa coste tardiness y earliness	Tardiness promedio	Earliness promedio	Ejemplares mejora coste total	Ejemplares mejora o iguala coste total
(0,1]	1134	4.3	-88.3%	494.7	57.6%	247.2	0.03%	247.5	271.0%	0.7	0.2	0.7%	73.4%
(1,2]	1134	11.0	-89.1%	2020.8	74.7%	603.1	0.08%	1417.7	155.8%	1.4	0.5	3.4%	34.4%
(2,3]	648	10.9	-88.1%	5446.4	45.0%	861.4	0.35%	4585.1	58.3%	3.5	1.4	7.3%	18.7%
(3,4]	324	9.1	-84.9%	13523.3	36.0%	1112.0	-0.01%	12411.3	40.6%	6.5	2.7	8.3%	11.7%
(4,5]	162	7.5	-80.2%	53188.6	70.4%	1262.3	0.11%	51926.4	73.4%	13.7	4.0	7.4%	9.9%
(5,6]	162	10.1	-78.2%	42162.9	10.9%	1547.1	0.80%	40615.8	11.3%	13.0	5.1	19.8%	20.4%
7	54	7.0	-74.8%	119217.9	53.4%	1585.3	0.92%	117632.6	54.5%	27.1	7.3	13.0%	13.0%
8	54	8.5	-74.2%	133999.9	66.1%	1755.6	0.85%	132244.3	67.5%	25.8	9.2	18.5%	18.5%
9	54	10.3	-71.6%	160036.8	64.1%	2005.9	0.53%	158030.8	65.5%	30.5	10.3	11.1%	11.1%
10	54	12.6	-70.4%	199883.3	26.2%	2316.3	0.53%	197567.0	26.6%	32.2	11.1	22.2%	22.2%
11	54	14.7	-70.1%	221644.5	27.7%	2552.3	1.21%	219092.2	28.1%	30.3	11.0	13.0%	13.0%
12	54	17.0	-70.9%	370385.7	79.7%	2800.6	0.57%	367585.2	80.8%	40.0	12.7	20.4%	20.4%
TOTAL	3888	8.7	-87.0%	23479.8	48.6%	782.1	0.33%	22697.7	51.2%	5.4	1.9	5.6%	38.1%

Tabla A3.3: Resultados promedio de Calibración General semidinámica comparada con HeufJSP(ζ).

Calibración Individual estática

Ratio n/m	Total ejemplares	Tiempo promedio por ejemplar [s]	Diferencia relativa del tiempo	Coste total promedio [€]	Diferencia relativa del coste total	Coste energía promedio [€]	Diferencia relativa coste energía	Coste tardiness y earliness promedio [€]	Diferencia relativa coste tardiness y earliness	Tardiness promedio	Earliness promedio	Ejemplares mejora coste total	Ejemplares mejora o iguala coste total
(0,1]	1134	237.3	545.7%	306.5	-2.3%	247.3	0.05%	59.2	-11.2%	0.3	0.2	15.1%	91.4%
(1,2]	1134	726.6	622.0%	1019.6	-11.9%	602.6	-0.01%	417.0	-24.7%	0.7	0.8	38.8%	76.5%
(2,3]	648	719.6	680.6%	3100.9	-17.4%	859.8	0.17%	2241.1	-22.6%	2.1	1.8	54.6%	72.7%
(3,4]	324	525.7	767.7%	8060.4	-18.9%	1111.1	-0.09%	6949.3	-21.3%	4.1	3.0	65.4%	70.4%
(4,5]	162	358.3	850.1%	24289.2	-22.2%	1257.3	-0.28%	23031.9	-23.1%	8.2	5.0	68.5%	71.6%
(5,6]	162	460.6	892.4%	27208.1	-28.5%	1525.9	-0.59%	25682.2	-29.6%	8.7	5.7	79.6%	80.9%
7	54	294.8	960.5%	77431.7	-0.3%	1568.5	-0.15%	75863.2	-0.3%	18.4	8.0	64.8%	64.8%
8	54	347.4	950.6%	63701.6	-21.0%	1745.2	0.26%	61956.4	-21.5%	16.7	9.4	77.8%	77.8%
9	54	401.9	1011.1%	87799.3	-10.0%	1994.3	-0.05%	85805.0	-10.2%	20.2	8.4	74.1%	74.1%
10	54	501.0	1082.6%	130468.2	-17.6%	2302.6	-0.07%	128165.6	-17.9%	23.1	8.9	66.7%	66.7%
11	54	555.5	1027.6%	121108.5	-30.2%	2517.6	-0.17%	118590.9	-30.7%	21.7	10.2	70.4%	70.4%
12	54	683.1	1067.2%	180010.6	-12.6%	2785.0	0.01%	177225.7	-12.8%	26.8	10.5	74.1%	74.1%
TOTAL	3888	517.7	667.8%	12894.9	-18.4%	779.1	-0.05%	12115.8	-19.3%	3.4	2.1	42.4%	79.2%

Tabla A3.4: Resultados promedio de Calibración Individual estática (IP^{Cle_1}) comparada con HeufJSP(ζ).

Ratio n/m	Total ejemplares	Tiempo promedio por ejemplar [s]	Diferencia relativa del tiempo	Coste total promedio [€]	Diferencia relativa del coste total	Coste energía promedio [€]	Diferencia relativa coste energía	Coste tardiness y earliness promedio [€]	Diferencia relativa coste tardiness y earliness	Tardiness promedio	Earliness promedio	Ejemplares mejora coste total	Ejemplares mejora o iguala coste total
(0,1]	1134	233.0	534.0%	310.6	-1.0%	247.3	0.05%	63.4	-5.0%	0.3	0.2	11.7%	91.9%
(1,2]	1134	712.8	608.3%	1156.5	0.0%	603.6	0.16%	552.9	-0.2%	0.8	0.7	31.9%	78.9%
(2,3]	648	712.8	673.2%	3194.8	-14.9%	858.5	0.01%	2336.3	-19.4%	2.1	1.8	45.2%	72.5%
(3,4]	324	510.1	742.1%	8517.3	-14.3%	1111.3	-0.07%	7405.9	-16.1%	4.4	3.2	54.6%	67.6%
(4,5]	162	333.6	784.7%	24413.2	-21.8%	1255.1	-0.46%	23158.1	-22.7%	8.7	4.7	58.6%	67.9%
(5,6]	162	433.1	833.2%	29263.0	-23.1%	1532.8	-0.14%	27730.2	-24.0%	9.2	5.9	68.5%	71.6%
7	54	273.6	884.1%	72891.1	-6.2%	1572.8	0.12%	71318.3	-6.3%	19.7	7.4	61.1%	64.8%
8	54	336.8	918.5%	76284.2	-5.4%	1742.4	0.10%	74541.8	-5.6%	17.2	10.7	74.1%	81.5%
9	54	393.2	987.1%	99152.3	1.7%	1988.6	-0.34%	97163.7	1.7%	20.1	9.8	70.4%	72.2%
10	54	454.4	972.5%	139256.3	-12.1%	2305.9	0.08%	136950.4	-12.3%	23.7	11.7	68.5%	70.4%
11	54	503.4	921.9%	135219.7	-22.1%	2516.1	-0.23%	132703.6	-22.4%	22.1	10.1	68.5%	72.2%
12	54	625.1	968.0%	228831.0	11.0%	2766.7	-0.64%	226064.4	11.2%	28.4	9.2	70.4%	70.4%
TOTAL	3888	505.1	649.0%	14346.0	-9.2%	779.1	-0.05%	13567.0	-9.7%	3.6	2.1	35.9%	79.3%

Tabla A3.5: Resultados promedio de Calibración Individual estática (IP^{Cle_2}) comparada con HeufJSP(ζ).

Ratio n/m	Total ejemplares	Tiempo promedio por ejemplar [s]	Diferencia relativa del tiempo	Coste total promedio [€]	Diferencia relativa del coste total	Coste energía promedio [€]	Diferencia relativa coste energía	Coste tardiness y earliness promedio [€]	Diferencia relativa coste tardiness y earliness	Tardiness promedio	Earliness promedio	Ejemplares mejora coste total	Ejemplares mejora o iguala coste total
(0,1]	1134	284.7	674.6%	337.1	7.4%	247.2	0.03%	89.9	34.8%	0.3	0.3	15.1%	93.9%
(1,2]	1134	875.1	769.5%	978.4	-15.4%	602.6	-0.01%	375.8	-32.2%	0.7	0.7	43.3%	84.9%
(2,3]	648	880.2	854.8%	2841.9	-24.3%	858.8	0.05%	1983.1	-31.5%	1.9	1.9	60.6%	82.3%
(3,4]	324	628.3	937.2%	8021.6	-19.3%	1107.0	-0.46%	6914.6	-21.7%	3.7	3.1	74.4%	79.9%
(4,5]	162	417.3	1006.5%	21994.6	-29.5%	1257.4	-0.27%	20737.2	-30.8%	7.6	4.5	80.9%	82.7%
(5,6]	162	532.6	1047.5%	25738.2	-32.3%	1531.6	-0.22%	24206.7	-33.7%	8.3	5.4	83.3%	84.6%
7	54	339.6	1121.5%	64763.1	-16.6%	1569.1	-0.12%	63194.0	-16.9%	17.2	8.7	74.1%	74.1%
8	54	408.7	1136.0%	66140.3	-18.0%	1737.4	-0.19%	64402.9	-18.4%	15.2	9.1	92.6%	92.6%
9	54	468.6	1195.5%	99901.2	2.5%	1993.7	-0.08%	97907.5	2.5%	19.5	8.7	81.5%	81.5%
10	54	570.3	1246.0%	113530.3	-28.3%	2301.9	-0.10%	111228.3	-28.7%	21.1	11.5	77.8%	77.8%
11	54	622.3	1163.2%	123024.0	-29.1%	2516.3	-0.22%	120507.7	-29.6%	19.7	10.2	85.2%	85.2%
12	54	777.2	1228.1%	150615.6	-26.9%	2679.9	-3.76%	147935.7	-27.2%	24.4	10.2	83.3%	83.3%
TOTAL	3888	621.1	821.2%	12062.0	-23.6%	776.7	-0.35%	11285.3	-24.8%	3.2	2.1	47.0%	86.4%

Tabla A3.6: Resultados promedio de Calibración Individual estática (IP^{Cle_3}) comparada con HeufJSP(ζ).

Calibración Individual semidinámica

Ratio n/m	Total ejemplares	Tiempo promedio por ejemplar [s]	Diferencia relativa del tiempo	Coste total promedio [€]	Diferencia relativa del coste total	Coste energía promedio [€]	Diferencia relativa coste energía	Coste tardiness y earliness promedio [€]	Diferencia relativa coste tardiness y earliness	Tardiness promedio	Earliness promedio	Ejemplares mejora coste total	Ejemplares mejora o iguala coste total
(0,1]	1134	629.9	1613.7%	328.9	4.8%	247.3	0.05%	81.6	22.3%	0.3	0.2	13.8%	95.0%
(1,2]	1134	1854.0	1742.1%	1056.4	-8.7%	601.8	-0.14%	454.6	-17.9%	0.7	0.6	41.8%	88.6%
(2,3]	648	1958.8	2024.7%	2860.4	-23.8%	858.8	0.05%	2001.6	-30.9%	1.9	1.8	57.6%	83.8%
(3,4]	324	1760.2	2805.8%	7404.9	-25.5%	1109.3	-0.25%	6295.5	-28.7%	4.0	2.8	74.7%	85.8%
(4,5]	162	1462.8	3779.0%	23504.9	-24.7%	1253.5	-0.58%	22251.4	-25.7%	7.9	4.6	77.8%	82.7%
(5,6]	162	2006.9	4223.7%	27023.8	-28.9%	1528.8	-0.39%	25495.0	-30.2%	8.4	5.4	81.5%	83.3%
7	54	1435.7	5064.8%	44008.0	-43.4%	1566.6	-0.28%	42441.4	-44.3%	16.0	8.9	79.6%	81.5%
8	54	1769.7	5251.6%	59386.4	-26.4%	1740.2	-0.03%	57646.1	-27.0%	15.5	8.1	92.6%	92.6%
9	54	2142.0	5822.5%	80386.7	-17.6%	1988.6	-0.34%	78398.1	-17.9%	17.5	8.6	87.0%	87.0%
10	54	2704.9	6284.3%	119018.1	-24.9%	2300.5	-0.16%	116717.6	-25.2%	22.0	12.3	79.6%	79.6%
11	54	3294.0	6586.5%	108567.0	-37.5%	2518.4	-0.14%	106048.7	-38.0%	19.5	8.3	81.5%	81.5%
12	54	3760.6	6325.7%	169155.8	-17.9%	2761.7	-0.82%	166394.0	-18.2%	25.6	9.9	92.6%	92.6%
TOTAL	3888	1552.0	2201.7%	11666.0	-26.1%	778.0	-0.19%	10888.0	-27.5%	3.2	2.0	45.8%	88.7%

Tabla A3.7: Resultados promedio de Calibración Individual semidinámica (IP^{CIs}) comparada con HeufJSP(ζ).

GRASP-HeufJSP

Ratio n/m	Total ejemplares	Tiempo promedio por ejemplar [s]	Diferencia relativa del tiempo	Coste total promedio [€]	Diferencia relativa del coste total	Coste energía promedio [€]	Diferencia relativa coste energía	Coste tardiness y earliness promedio [€]	Diferencia relativa coste tardiness y earliness	Tardiness promedio	Earliness promedio	Ejemplares mejora coste total	Ejemplares mejora o iguala coste total
(0,1]	1134	390.5	962.4%	249.3	-20.6%	245.8	-0.52%	3.5	-94.8%	0.0	0.2	30.4%	99.7%
(1,2]	1134	480.1	377.1%	698.5	-39.6%	600.1	-0.43%	98.5	-82.2%	0.2	0.6	64.1%	87.3%
(2,3]	648	483.6	424.5%	1995.9	-46.8%	857.7	-0.08%	1138.3	-60.7%	1.1	1.7	71.9%	77.0%
(3,4]	324	483.5	698.2%	5733.0	-42.3%	1111.1	-0.09%	4621.9	-47.7%	3.1	3.2	70.1%	70.1%
(4,5]	162	482.8	1180.1%	19637.5	-37.1%	1255.5	-0.42%	18382.0	-38.6%	6.8	5.3	74.1%	74.1%
(5,6]	162	482.9	940.3%	36531.9	-4.0%	1530.1	-0.31%	35001.9	-4.1%	9.4	6.5	56.8%	56.8%
7	54	482.1	1634.4%	72758.7	-6.4%	1566.7	-0.27%	71192.0	-6.5%	18.9	9.9	63.0%	63.0%
8	54	482.5	1359.1%	108335.5	34.3%	1744.7	0.23%	106590.9	35.0%	18.7	12.0	51.9%	51.9%
9	54	482.6	1234.4%	264314.9	171.1%	1994.3	-0.05%	262320.6	174.6%	26.1	10.4	20.4%	20.4%
10	54	482.7	1039.3%	396610.2	150.4%	2296.3	-0.34%	394313.9	152.6%	31.9	13.0	13.0%	13.0%
11	54	483.7	881.8%	342372.3	97.2%	2494.5	-1.08%	339877.8	98.7%	30.9	12.5	3.7%	3.7%
12	54	484.8	728.5%	623271.4	202.5%	2756.0	-1.03%	620515.4	205.2%	41.7	16.4	7.4%	7.4%
TOTAL	3888	455.3	575.3%	28533.7	80.6%	776.8	-0.34%	27756.8	84.8%	3.5	2.3	53.1%	80.9%

Tabla A3.8: Resultados promedio de GRASP-HeufJSP con azar orientado comparado con HeufJSP(ζ).

Ratio n/m	Total ejemplares	Tiempo promedio por ejemplar [s]	Diferencia relativa del tiempo	Coste total promedio [€]	Diferencia relativa del coste total	Coste energía promedio [€]	Diferencia relativa coste energía	Coste tardiness y earliness promedio [€]	Diferencia relativa coste tardiness y earliness	Tardiness promedio	Earliness promedio	Ejemplares mejora coste total	Ejemplares mejora o iguala coste total
(0,1]	1134	391.2	964.3%	249.3	-20.6%	246.1	-0.43%	3.3	-95.1%	0.0	0.2	30.2%	99.6%
(1,2]	1134	480.2	377.1%	697.2	-39.7%	600.4	-0.37%	96.7	-82.5%	0.2	0.6	62.9%	85.4%
(2,3]	648	484.7	425.7%	1945.7	-48.2%	857.8	-0.07%	1087.9	-62.4%	1.2	1.6	66.8%	71.1%
(3,4]	324	483.3	697.9%	5034.9	-49.4%	1110.6	-0.14%	3924.3	-55.6%	2.9	3.2	68.2%	68.2%
(4,5]	162	482.5	1179.4%	22145.2	-29.0%	1249.7	-0.88%	20895.5	-30.2%	6.7	5.5	69.8%	69.8%
(5,6]	162	483.1	940.8%	29046.9	-23.6%	1531.9	-0.19%	27515.0	-24.6%	8.7	6.3	59.3%	59.3%
7	54	482.2	1634.8%	73723.7	-5.1%	1569.0	-0.12%	72154.7	-5.2%	16.6	8.1	72.2%	72.2%
8	54	482.1	1357.9%	95998.4	19.0%	1739.2	-0.09%	94259.2	19.4%	17.6	10.8	57.4%	57.4%
9	54	482.9	1235.2%	187992.5	92.8%	1982.0	-0.67%	186010.5	94.7%	22.7	11.9	33.3%	33.3%
10	54	482.4	1038.6%	292450.5	84.6%	2295.7	-0.37%	290154.8	85.9%	28.7	13.1	20.4%	20.4%
11	54	483.1	880.7%	320776.8	84.8%	2517.5	-0.17%	318259.3	86.0%	29.7	13.9	11.1%	11.1%
12	54	483.8	726.6%	498022.4	141.7%	2760.1	-0.88%	495262.3	143.6%	38.0	13.1	11.1%	11.1%
TOTAL	3888	455.7	575.8%	23555.2	49.1%	777.0	-0.32%	22778.2	51.7%	3.3	2.3	52.2%	79.7%

Tabla A3.9: Resultados promedio de GRASP-HeufJSP con azar no orientado comparado con HeufJSP(ζ).

SA-HeufJSP

Ratio n/m	Total ejemplares	Tiempo promedio por ejemplar [s]	Diferencia relativa del tiempo	Coste total promedio [€]	Diferencia relativa del coste total	Coste energía promedio [€]	Diferencia relativa coste energía	Coste tardiness y earliness promedio [€]	Diferencia relativa coste tardiness y earliness	Tardiness promedio	Earliness promedio	Ejemplares mejora coste total	Ejemplares mejora o iguala coste total
(0,1]	1134	483.2	1214.7%	250.0	-20.4%	246.4	-0.28%	3.5	-94.7%	0.1	0.2	25.6%	94.8%
(1,2]	1134	488.4	385.3%	693.6	-40.0%	605.0	0.40%	88.6	-84.0%	0.3	0.8	36.2%	63.4%
(2,3]	648	487.8	429.1%	1852.3	-50.7%	864.3	0.69%	988.0	-65.9%	1.2	2.2	44.4%	53.5%
(3,4]	324	485.4	701.3%	4787.1	-51.8%	1119.3	0.64%	3667.8	-58.5%	2.9	3.5	53.1%	54.3%
(4,5]	162	483.5	1182.2%	20964.0	-32.8%	1265.0	0.33%	19699.0	-34.2%	7.1	5.3	57.4%	58.6%
(5,6]	162	484.4	943.7%	23388.8	-38.5%	1540.0	0.34%	21848.8	-40.1%	7.9	6.5	61.7%	62.3%
7	54	483.1	1637.7%	62349.4	-19.7%	1578.3	0.47%	60771.1	-20.2%	15.2	9.2	64.8%	66.7%
8	54	483.3	1361.6%	53376.3	-33.8%	1754.8	0.81%	51621.5	-34.6%	13.3	11.3	83.3%	83.3%
9	54	483.5	1236.9%	94432.5	-3.2%	1998.5	0.16%	92434.0	-3.2%	19.5	10.4	66.7%	66.7%
10	54	484.0	1042.5%	105834.0	-33.2%	2312.5	0.36%	103521.5	-33.7%	20.0	13.9	61.1%	61.1%
11	54	485.2	884.9%	129324.3	-25.5%	2520.4	-0.06%	126803.9	-25.9%	20.5	11.8	74.1%	74.1%
12	54	486.2	730.7%	199931.5	-3.0%	2795.7	0.40%	197135.8	-3.0%	27.4	14.9	64.8%	64.8%
TOTAL	3888	485.8	620.5%	11792.7	-25.3%	782.5	0.39%	11010.1	-26.7%	2.8	2.4	40.6%	70.4%

Tabla A3.10: Resultados promedio de SA-HeufJSP comparado con HeufJSP(ζ).

GRASPSA-HeufJSP

Ratio n/m	Total ejemplares	Tiempo promedio por ejemplar [s]	Diferencia relativa del tiempo	Coste total promedio [€]	Diferencia relativa del coste total	Coste energía promedio [€]	Diferencia relativa coste energía	Coste tardiness y earliness promedio [€]	Diferencia relativa coste tardiness y earliness	Tardiness promedio	Earliness promedio	Ejemplares mejora coste total	Ejemplares mejora o iguala coste total
(0,1]	1134	482.2	1211.9%	249.3	-20.6%	246.1	-0.43%	3.2	-95.1%	0.0	0.2	29.9%	99.5%
(1,2]	1134	486.7	383.6%	672.7	-41.9%	600.8	-0.30%	71.8	-87.0%	0.2	0.7	61.5%	86.2%
(2,3]	648	486.4	427.5%	1757.0	-53.2%	858.9	0.06%	898.1	-69.0%	1.0	1.8	71.5%	77.9%
(3,4]	324	483.7	698.5%	4250.8	-57.2%	1113.3	0.10%	3137.5	-64.5%	2.5	3.1	79.6%	79.9%
(4,5]	162	482.7	1179.9%	15326.8	-50.9%	1258.0	-0.22%	14068.8	-53.0%	5.7	5.8	82.7%	82.7%
(5,6]	162	485.0	944.8%	18943.9	-50.2%	1536.8	0.12%	17407.1	-52.3%	6.5	6.2	82.1%	82.1%
7	54	482.2	1634.8%	49638.7	-36.1%	1567.3	-0.23%	48071.4	-36.9%	14.9	8.5	85.2%	85.2%
8	54	482.6	1359.3%	58813.2	-27.1%	1744.4	0.21%	57068.8	-27.7%	13.9	13.2	83.3%	83.3%
9	54	482.8	1234.9%	100824.7	3.4%	1986.7	-0.43%	98838.0	3.5%	18.1	11.3	74.1%	74.1%
10	54	483.0	1040.0%	132005.0	-16.7%	2295.5	-0.38%	129709.5	-16.9%	21.4	11.7	75.9%	75.9%
11	54	483.3	881.1%	122167.0	-29.6%	2509.6	-0.49%	119657.5	-30.1%	18.7	11.8	79.6%	79.6%
12	54	483.9	726.9%	256647.2	24.5%	2778.1	-0.23%	253869.1	24.9%	29.0	12.8	68.5%	68.5%
TOTAL	3888	484.5	618.6%	12345.2	-21.8%	778.3	-0.15%	11567.0	-23.0%	2.6	2.3	58.5%	87.2%

Tabla A3.11: Resultados promedio de GRASPSA-HeufJSP comparado con HeufJSP(ζ).